ASPAI
MALLORCA • 2026

Advances in Signal Processing and Artificial Intelligence

Proceedings of the 8th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2026)

Edited by Sergey Y. Yurish

IFSA

ASPAI
MALLORCA • 2026

Advances in Signal Processing and Artificial Intelligence:

Proceedings of the 8th International Conference on Advances in Signal Processing and Artificial Intelligence

24-26 June 2026
Palma de Mallorca (Balearic Islands), Spain

Edited by Sergey Y. Yurish

IFSA Publishing

Sergey Y. Yurish, *Editor*
Advances in Signal Processing and Artificial Intelligence
ASPAI' 2026 Conference Proceedings

E-mail (for orders and customer service enquires): ifsa.books@sensorsportal.com

Visit our Home Page on http://www.sensorsportal.com

ISBN: 978-84-09-86145-3
BN-20260619-XX
BIC: UYQ

Contents

Foreword

Signal processing and artificial intelligence have become deeply interconnected scientific and technological fields. Signal processing provides the theoretical and practical foundations for acquiring, analysing, interpreting and transforming data from the physical world, while artificial intelligence offers powerful methods for learning from such data, recognizing patterns, making predictions, supporting decisions and enabling autonomous behaviour. Together, these disciplines are shaping many of the most dynamic areas of modern engineering, computer science, sensing technologies, robotics, communications, medical and industrial applications.

The *8th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2026)* has been organized as an international forum for researchers, engineers, developers and practitioners to present recent advances, exchange ideas and discuss emerging challenges at the intersection of signal processing and artificial intelligence. The papers included in these Conference Proceedings reflect the broad and interdisciplinary character of the conference. They address both theoretical developments and practical implementations, including real-time computer vision systems, analog and digital signal processing, graph convolutional networks, Kalman filtering and recurrent neural models, meta-learning for sonar data analysis, image clustering, multimodal co-simulation platforms, secure AI architectures, acoustic feature selection for biomedical applications, embedded deep learning, point cloud registration, computational complexity, pseudo-expert systems for engineering materials, CT surface reconstruction, fuzzy-neural control of industrial robots and classification of targets from multi-view sonar imagery.

A particularly important feature of the present volume is its attention to the transition from algorithms to reliable systems. Many contributions demonstrate that progress in artificial intelligence is not achieved by models alone, but through careful integration of data acquisition, preprocessing, modelling, optimization, validation, deployment and interpretation. This is especially relevant for applications where systems must operate under real-world constraints, such as limited data availability, computational restrictions, noisy measurements, strict latency requirements, complex physical environments or safety-critical decision-making conditions.

The conference also highlights the growing importance of robustness, efficiency and applicability. Modern AI-based systems must not only achieve high accuracy in controlled experiments, but also maintain stable performance when transferred to practical environments. Whether the task concerns real-time human action recognition, sonar-based object classification, industrial robot control, medical signal analysis, image registration or structural engineering assessment, the same fundamental questions arise: how can we build systems that are accurate, explainable, computationally feasible and trustworthy.

These Conference Proceedings show that the future of signal processing and artificial intelligence lies in close cooperation between theory and practice. Mathematical models, learning algorithms, sensors, embedded platforms, simulation environments and domain-specific expertise must work together. Such cooperation is essential for creating technologies that are not only innovative, but also useful, reliable and ready to address real engineering and scientific problems.

On behalf of the Organizing Committee, I would like to express my sincere gratitude to all authors for their valuable contributions and for sharing their latest research results with the international community. I also extend my appreciation to the reviewers for their careful evaluations and constructive comments, which helped to maintain the scientific quality of the conference. Special thanks are due to the members of the International Program Committee, invited speakers, session chairs and all colleagues who contributed to the preparation and successful organization of ASPAI' 2026.

I hope that this volume will serve not only as a record of the scientific work presented at the conference, but also as a source of inspiration for future research, collaboration and innovation. May the ideas presented here contribute to further advances in signal processing and artificial intelligence, and to their responsible application in science, industry and society.

Prof., Dr. Sergey Y. Yurish
ASPAI' 2026 Conference Chairman

(003)

Best Practices for Implementing Real-Time Computer Vision Systems

Hannes Fassold and Stefanie Onsori-Wechtitsch
Joanneum Research – Digital, Steyrergasse 17, 8010 Graz, Austria
E-mail: <givenname>.<surname>@joanneum.at

Summary: Implementing real-time computer vision systems requires more than just a powerful neural network; it demands a highly cohesive architecture where every component operates seamlessly without causing delays. Because practical pipelines rely on interconnected stages (such as video input, data preprocessing, and network communication) a single bottleneck will instantly destroy the real-time capabilities of the entire system. This work offers a practical roadmap to navigate these real-world bottlenecks. Essential optimization strategies for individual tasks are outlined, including the use of high-performance tensor libraries, just-in-time compilation, and model enhancements like quantization. We provide information on how to leverage asynchronous and concurrent processing to maintain overall system responsiveness. Multithreading can be employed to efficiently handle I/O-bound operations, such as video streaming and network requests. Simultaneously, multiprocessing can be utilized to spawn independent sub-processes, achieving true concurrent CPU and GPU execution for computationally expensive neural network tasks. These architectural best practices are showcased through an application performing real-time human action recognition for virtual humans in VR.

Keywords: Real-time processing, Machine learning, Computer vision, Parallel processing.

1. Introduction

We expect real-time computer vision systems to see and react instantly, whether that’s a self-driving car spotting a pedestrian or a robotic arm identifying a defect on a fast-moving assembly line. However, as any engineer who has deployed these systems knows, getting a model to perform well in a Python notebook is very different from making it run smoothly in the real world. The challenge is no longer just about building the smartest model; it is about building a system that is fast, efficient and reliable. This work is designed to be a brief and practical roadmap for that journey. We move past the theoretical hype to look at the "nuts and bolts" of what it actually takes to build computer vision systems which run in real time.

It is important to keep in mind that a practical computer vision system comprises not only a single neural network. Usually, it contains a variety of components, like components for reading the input (e.g. one or multiple video streams), components for preprocessing the content, multiple neural networks which do different tasks (like object detection, pose estimation, tracking) and components for displaying the result or sending it via network (e.g. via REST). The challenge now is that if *only one* of these components is not real-time capable (e.g. because the latency of the REST calls is too high), this will instantly destroy the real-time capability of the *whole* computer vision system.

Despite its practical importance, there are just a few works in the literature focusing specifically on techniques for real-time computer vision systems. In [1] a programming model and architectural design is proposed for real-time machine learning. The work of [2] provides techniques for low-latency inference of large language models. Finally, the work of [3] gives an overview of model optimization for edge deployment.

Achieving real-time performance for the whole system is usually only possible by optimizing the components itself in various ways (see Section 2) in combination with asynchronous processing (see Section 3) of multiple components. In Section 4, we will briefly describe a system for real-time human action recognition in VR built upon these principles. We will focus on the *Python* language and a system which utilizes both CPU and GPU, as this is the de-facto standard for machine learning applications.

2. General Optimization Strategies

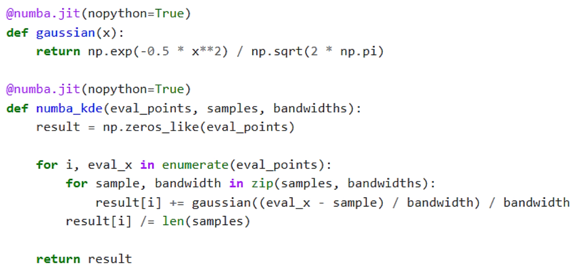
In this section, we will focus on strategies for optimizing the runtime of individual components via a variety of techniques, both for general processing steps (e.g. the preprocessing of the input images) as well as for neural network inference. As a first step always, a profiling should be done in order to identify the computationally expensive operations, naturally the focus should be put on these.

2.1. General Processing Steps

In the following, we will describe several important techniques for runtime optimization which can be applied individually or in combination.

Loop iterations over large multidimensional arrays are very expensive in Python due to multiple reasons (large overhead of Python interpreter, no loop vectorization or parallelization etc.). Therefore, one should aim for replacing these loops with a single function call (or a combination of calls) from a high-performance matrix / tensor library like *numexpr*, *NumPy*, *JAX, CuPy* or *PyTorch*.

Another way to speed up python code significantly is via *just-in-time compilation* (JIT). Libraries like *Numba*[1] (see Fig. 1), *JAX* and *Taichi Lang*[2] translate Python code into optimized machine code using compiler frameworks like LLVM or XLA. This process bypasses the Python interpreter overhead and enables optimizations such as SIMD vectorization and parallelization on CPUs as well as GPUs.



```python
@numba.jit(nopython=True)
def gaussian(x):
    return np.exp(-0.5 * x**2) / np.sqrt(2 * np.pi)

@numba.jit(nopython=True)
def numba_kde(eval_points, samples, bandwidths):
    result = np.zeros_like(eval_points)

    for i, eval_x in enumerate(eval_points):
        for sample, bandwidth in zip(samples, bandwidths):
            result[i] += gaussian((eval_x - sample) / bandwidth) / bandwidth
        result[i] /= len(samples)

    return result
```

Fig. 1. Example implementation of 1D Gaussian Kernel estimator in Numba.

Porting runtime-critical routines to a compiled language and invoking them via a Python wrapper can also bring significant speedups, often by an order of magnitude. The *C/C++* programming language is usually employed for that in combination with a Python wrapper library like *pybind11*[3] or *nanobind*. The Rust programming language is also a good alternative in combination with the Python wrapper libraries *PyO3*[4], *rust-numpy* and *maturin*.

```cpp
py::class_<Jrs::Cuda::ImageProps>(m, "ImageProps")
    .def(py::init<>())
    .def(py::init<std::string>())
    .def("getPropString", &Jrs::Cuda::ImageProps::getPropString)
    .def("setFromPropString", &Jrs::Cuda::ImageProps::setFromPropString)
    .def_readwrite("width", &Jrs::Cuda::ImageProps::width)
    .def_readwrite("height", &Jrs::Cuda::ImageProps::height)
    .def_readwrite("nChannels", &Jrs::Cuda::ImageProps::nChannels)
    .def_readwrite("depth", &Jrs::Cuda::ImageProps::depth)
```

Fig. 2. Python wrapper for a C++ class with the *pybind11* wrapper library.

Finally, there are often faster alternatives available for popular Python libraries with an identical or similar interface. For example for handling data frames, high-performance libraries like *Polars*[5], *cuDF*, *Modin* and *Dask* can be employed instead of *Pandas*. Similarly, for web requests the libraries *niquests* and *httpx* are often significantly faster than the *requests* library.

2.2. Neural Network Inference

A range of well-established optimization techniques are commonly used to reduce the cost of neural network inference. One widely adopted approach is *model distillation*, where a smaller *student network* is trained to mimic the behavior of a larger, more complex *teacher model*. Another key technique is *quantization*, which lowers both memory usage and inference latency by representing 32-bit floating-point weights with lower-precision formats such as 16-bit, 8-bit or even 4-bit integers.

The performance can be further improved by taking advantage of highly optimized GPU inference frameworks like *NVIDIA TensorRT*[6] or specialized frameworks for large language models and vision-language models such as *vLLM*[7], *lmdeploy* and *SGLang*. Many neural network architectures also offer multiple backbone variants of varying complexity and support different input resolutions. For example, the very popular *Swin Transformer* backbone supports a variable input resolution and comes in four different variants: *tiny*, *small*, *base* and *large*. Selecting the smallest input resolution and backbone variant that still gives the desired quality for the task can give a significant speedup for the inference.

3. Asynchronous / Concurrent Processing

Asynchronous and concurrent processing allows an application to process multiple of its components in parallel, instead of executing everything one step after another. This is crucial for the real-time capability of a computer vision system, especially when it has to deal with slow I/O operations or computationally expensive tasks. In Python, this is usually handled in two main ways: *multithreading*[8] and *multiprocessing*[9]. In the following, we will describe both concepts briefly.

3.1. Multithreading

In Python, multithreading allows a program to handle multiple tasks concurrently in a single process by switching between different threads of execution, using constructs like *Thread* and *asyncio*. It is important to keep in mind that multi-threading in Python cannot be regarded as true parallelism for computationally expensive tasks because the *Global Interpreter Lock* (GIL) demands that only one thread executes Python bytecode at a time, effectively serializing execution.

Furthermore, multiple concurrent GPU inference calls with PyTorch usually leads to *implicit*

---

[1] https://numba.pydata.org/
[2] https://www.taichi-lang.org/
[3] https://github.com/pybind/pybind11
[4] https://github.com/pyo3/pyo3
[5] https://github.com/pola-rs/polars
[6] https://tensorrt.org/
[7] https://github.com/vllm-project/vllm
[8] https://docs.python.org/3/library/threading.html
[9] https://docs.python.org/3/library/multiprocessing.html

serialization, as all threads likely employ the *same (default) CUDA stream* for GPU kernel execution, effectively queuing operations on the GPU instead of running them in parallel. Consequently, multithreading is *best suited for I/O-bound operations*, such as streaming video input or sending results via a REST API, where the component would otherwise sit idle waiting for data to receive or send out.

3.2. Multiprocessing

Unlike multithreading, which as mentioned is constrained by the Global Interpreter Lock (GIL), the approach via *multiprocessing* involves spawning independent sub-processes, each equipped with its own dedicated Python interpreter and memory space. This decoupling ensures that multiple components can execute *truly concurrently* for both CPU and GPU operations. On the other hand, the creation (spawning) of a new sub-process is a heavyweight operation, resulting in non-trivial startup latency compared to the creation of a thread. Furthermore, since processes do not share a common address space, inter-process communication (IPC) is needed in combination with serialization of Python objects via *pickling*. Especially for large objects like images, this can induce significant overhead.

4. Real-Time Human Action Recognition

A computer vision system implemented according to these guidelines is presented in [4, 5]. The *Visual Analyzer* Python application is able to detect multiple actions performed by virtual humans in VR in real time.

The application does the processing of the input video stream (generated by Unity) in multiple steps. First, all virtual humans are detected with a deep learning based object detector (*Yolo-V4*) and tracked with an optical flow based method. In parallel, for all detected humans their 2D pose (skeleton) is calculated with a deep learning based pose estimation algorithm (*RTMpose*). The actions of all virtual humans are now calculated with a deep learning based action recognition algorithm (*PoseConv3D*) from the trajectory of its 2D poses within an analysis period of roughly two seconds. All detected actions are then sent back via REST API to the PC where Unity runs. Initial experiments on several XR Unity scenes recordings with different types of virtual humans demonstrate that their actions can be detected robustly, as can be seen in Fig. 3.

The application has been optimized heavily with the methods presented in this work, in order to make it real-time capable. Multiprocessing is employed for the AI components which are responsible for object detection and tracking, pose estimation and action classification. In contrast, multithreading is used for video I/O and sending the result back to Unity via REST API. The neural network inference for the pose estimation step has been ported from PyTorch to TensorRT, which makes it four times faster.

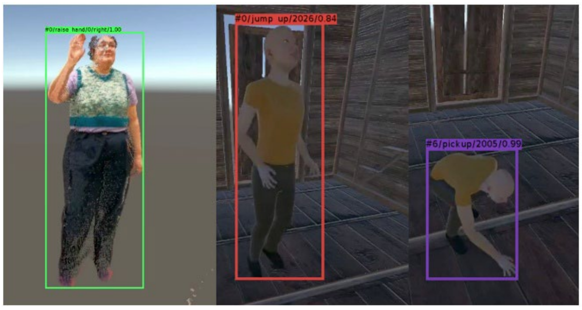


Fig. 3. Illustration of successfully detected actions for different kinds of virtual humans.

With all these optimizations, the application is able to detect human actions in real-time for up to five humans in the scene, with a latency of roughly two seconds.

5. Conclusion

We described a variety of practical methods for achieving real-time performance for computer vision systems, ranging from general optimization strategies to asynchronous / concurrent processing and applied these methods successfully to an application for real-time human action recognition.

Acknowledgements

This work has received funding from the European Union's Horizon Europe research and innovation programme under grant No. 101135025, "Presence XR".

References

[1]. R. Nishihara, et al., Real-time machine learning: The missing pieces, in *Proceedings of the* 16^{th} *Workshop on Hot Topics in Operating Systems (HotOS)*, 2017, pp. 106-110.

[2]. Z. Chen, et al., Network edge inference for large language models: Principles, techniques, and opportunities, *ACM Computing Surveys*, 2026 (in print).

[3]. X. Wang, et al., Optimizing edge AI: A comprehensive survey on data, model, and system strategies, *arXiv*, 2025, arXiv:2501.03265.

[4]. H. Fassold, I see what you do: The HumAct algorithm for live human action recognition and applications, in *Proceedings of the* 13^{th} *International Symposium on Computing and Networking Workshops (CANDARW)*, 2025, pp. 128-134.

[5]. H. Fassold, Enabling natural interactions with virtual humans in XR via real-time human action recognition, in *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2026.

(004)

Sensitivity Optimization of 4th-Order Bandpass Filter in CBQ Structure

L. Mikša [1], N. Stojković [1], Z. Šverko [1] and Sanja Grbac Babić [2]
[1] University of Rijeka, Faculty of Engineering, Vukovarska 58, 51000 Rijeka, Croatia
[2] Istarsko Veleučilište – Università Istriana di Scienze Applicate, Riva 6, 52100 Pula, Croatia
Tel.: + 385 51 651 445 / +385 51 505 720
E-mail: zoran.sverko@riteh.uniri.hr

Summary: In this paper, four different analog filter designs with identical specifications are presented. The key quality metric analyzed is sensitivity, defined as a change in frequency response with respect to change in element values due to component aging, external factors, tolerances, etc. Sensitivity is compared for cascade structure (CAS), negative-feedback structure also known as cascade of biquads (CBQ), and two different sensitivity-optimized CBQ structures. Sensitivity analysis is performed in MATLAB, while the Monte Carlo analysis is carried out in LTspice. As an optimization constraint, the ratio of element values is used. One sensitivity-optimized design is done with maximal ratio on the defined domain, and the other with optimally minimized ratio. As a result of optimization, the optimal design of sensitivity-optimized filter is proposed with specified central frequency of 4 kHz and normalized bandwidth of 0.25. It is shown that CBQ structure inherently reduces the sensitivity, and with further optimization, provides minimal sensitivity of the given analog filter.

Keywords: Filter design, Sensitivity optimization, Schoeffler sensitivity, Monte Carlo analysis.

1. Introduction

Despite the rapid development and widespread use of digital filters over the past few decades, analog filters are still widely used in areas such as measurement, instrumentation, audio and front-end signal conditioning [1-4]. While digital filters are reconfigurable, have memory and are less sensitive to noise and component aging, analog filters are still often preferred in high-frequency, high-precision, and low-latency applications due to ADC/DAC limitations and anti-aliasing requirements.

The performance of an analog filter is reflected in meeting the desired specifications, but also in robustness against environmental variations, such as mechanical stress, vibrations, temperature changes, and aging. Among the key quality metrics of an analog filter are noise and sensitivity. This paper analyzes the sensitivity of three different structures of the same fourth-order bandpass filter: a cascaded structure (CAS), negative-feedback structure, also known as cascade of biquads (CBQ) and an optimized CBQ design, which is expected to have the lowest sensitivity through the proposed optimization. Some previous optimizations are done in [5, 6] but optimization of CBQ design for BP is performed here. Similar optimization for LP filter is done in [7]. Finally, a practical tradeoff will be proposed.

2. Filter Realizations

Second-order bandpass filter sections are implemented using the Single Amplifier Biquad (SAB) topology, as shown in Fig. 1. Specified central frequency is 4 kHz while the normalized bandwidth is 0.25. The transfer function approximation is Chebyshev Type 1; passband ripple is 0.1 dB.

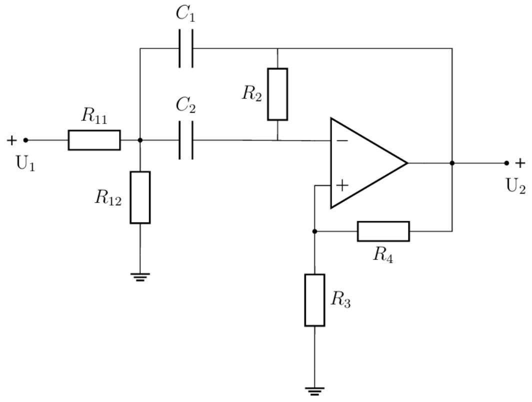


Fig. 1. Single Amplifier Biquad filter section (2nd-order).

The transfer function of the filter section shown in Fig. 1 is

$$T_{BP}(s) = -\frac{\left(1+\frac{R_3}{R_4}\right)\frac{1}{R_{11}C_1}s}{s^2+\left(\frac{1}{R_2C_1}+\frac{1}{R_2C_2}-\frac{R_3}{R_1R_4C_1}\right)s+\frac{1}{R_1R_2C_1C_2}}, \quad (1)$$

where R_1 is parallel resistances of R_{11} and R_{12}.

Using precalculated coefficients from the corresponding Chebyshev table (for the ripple of 0.1 dB), normalized element values can be calculated by comparing the normalized transfer function with the expressions for filter parameters obtained by comparison of Eq. (1) with the general form of a second-order bandpass transfer function

$$T_{BP}(s) = k\frac{\frac{\omega_p}{Q_p}s}{s^2+\frac{\omega_p}{Q_p}s+{\omega_p}^2} \quad (2)$$

The expressions for filter parameters (gain, Q-factor, and pole frequency, respectively) are

$$k = \frac{\frac{C_2}{R_{11}}\left(\frac{1}{R_3}+\frac{1}{R_4}\right)}{\frac{C_1+C_2}{R_2R_3}-\frac{C_2}{R_1R_4}}, \tag{3}$$

$$Q = \frac{\sqrt{\frac{R_2C_1}{R_1C_2}}}{1+\frac{C_1}{C_2}-\frac{R_2R_3}{R_1R_4}}, \tag{4}$$

$$\omega_p = \sqrt{\frac{1}{R_1R_2C_1C_2}} \tag{5}$$

Denormalization of element values is done on central frequency f_0 = 4 kHz with the chosen capacitor value of 10 nF.

Similar procedure applies for the CBQ structure, with the exception of a differential amplifier used as a negative feedback which is defined with Eq. (6) and Eq. (7). The circuit diagram of the CBQ structure is shown in Fig. 11.

$$\frac{R_{10}}{R_{30}} = \frac{R_{20}}{R_{40}}, \tag{6}$$

$$\beta = \frac{R_{30}}{R_{40}} \tag{7}$$

Table 1 shows the calculated element values for CAS and CBQ filter structures. It can be observed that the element values of the 2nd order sections within the CBQ structure are identical, which significantly simplifies the optimization procedure. This stems from the fact that the parameters of the individual filter sections are also identical.

3. Sensitivity Analysis

A method for sensitivity analysis is given in [5-8] and it is defined as a derivative of frequency response with respect to the individual element, Eq. (8).

$$S_{x_i}^{|T(j\omega)|} = \frac{d|T(j\omega)|}{dx_i}\frac{x_i}{|T(j\omega)|} \tag{8}$$

Eq. (8) can be interpreted as a rate of change of frequency response with respect to variations in each element. These variations happen due to already mentioned causes and the objective is to minimize their impact.

The Schoeffler sensitivity function is defined as a sum of squares of sensitivities of each individual element as shown in Eq. (9), and multiparameter sensitivity is defined with Eq. (10).

$$I_s^{\,2}(\omega) = \sum_i\left(S_{x_i}^{|T(j\omega)|}\right)^2, \tag{9}$$

$$M = \int_{\omega_2}^{\omega_2} I_s^{\,2}(\omega)d\omega \tag{10}$$

Fig. 2 shows comparison of total sensitivity between CAS and CBQ structures and it can be seen, as expected, CBQ has lower sensitivity. Furthermore, Fig. 3 shows that the element with the biggest contribution to the total sensitivity is the resistor R_2; thus, it is intention to minimize its sensitivity in order to minimize the total sensitivity as well.

Table 1. Element values of CAS and CBQ filter structures.

Element	**CAS**	**CBQ**	**SO CBQ, k = 5**	**SO CBQ, k = 10**
$R_{10}[\Omega]$	/	2326	2326	2326
$R_{20}[\Omega]$	/	3979	3979	3979
$R_{30}[\Omega]$	/	2326	2326	2326
$R_{40}[\Omega]$	/	3979	3979	3979
$R_{111}[\Omega]$	48656	48666	35426	33747
$R_{121}[\Omega]$	3934	4333	4482	4511
$R_{21}[\Omega]$	3640	3979	19895	39789
$R_{31}[\Omega]$	7353	7351	4168	3770
$R_{41}[\Omega]$	3979	3979	3979	3979
$C_{11}[nF]$	10	10	10	10
$C_{21}[nF]$	10	10	2	1
$R_{112}[\Omega]$	48663	48666	35426	33747
$R_{122}[\Omega]$	4776	4333	4482	4511
$R_{22}[\Omega]$	4349	3979	19895	39789
$R_{32}[\Omega]$	7353	7351	4168	3770
$R_{42}[\Omega]$	3979	3979	3979	3979
$C_{12}[nF]$	10	10	10	10
$C_{22}[nF]$	10	10	2	1

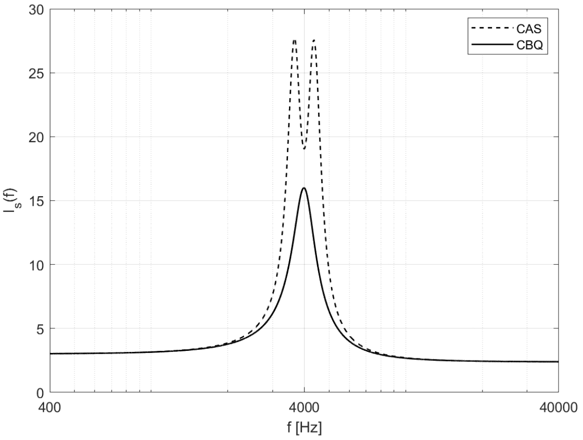


Fig. 2. Comparison of CAS and CBQ total sensitivities.

This minimization can be achieved by introducing an additional parameter *k*, defined as a ratio of resistors R_2 and R_1. In order to keep the pole frequency unchanged, the value of *k* is also equal to the ratio of capacitors C_1 and C_2, as shown with Eq. (11) and Eq. (12)

$$R_2 = kR_1, \tag{11}$$

$$C_2 = \frac{1}{k}C_1, \tag{12}$$

so that ω_p defined by Eq. (5) remains constant. This way, the optimization constraint is the ratio between element values. Namely, larger k can cause various problems in practical implementation of the filter, such as uneven aging of the components due to different manufacturing technologies, larger noise contribution, parasitics, etc.

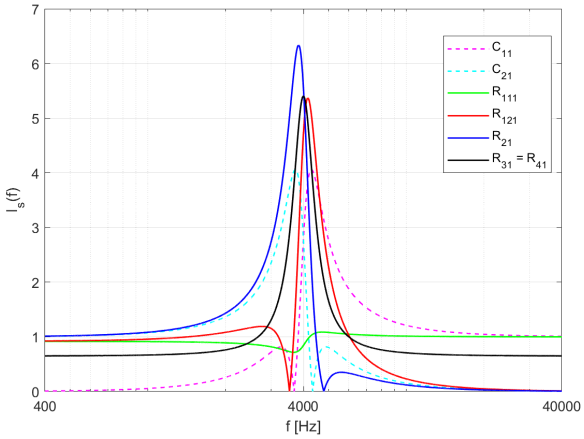


Fig. 3. Sensitivity curves of individual elements (CBQ).

The initial domain of k is defined with

$$k \in [0.1, 10], \tag{13}$$

and swept by recording sensitivity on a fine grid. The scalar value used to quantify sensitivity is defined by Eq. (10) which represents area under the curve of total sensitivity at each point; for example, area under the curve on Fig. 2. The results are shown on Fig. 4, and it can be seen that the lowest multiparameter sensitivity is lying on the very edge of the defined domain, specifically for $k = 10$.

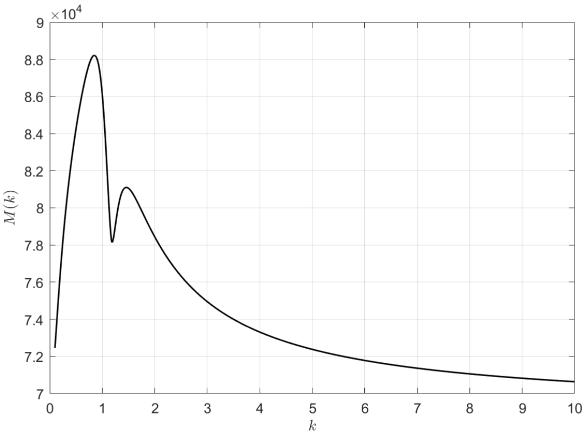


Fig. 4. Multiparameter sensitivity function of the CBQ structure.

The element values were recalculated and are shown in Table 1. Sensitivity curve for this case, alongside the curves for cascade and negative-feedback structure are shown on Fig. 5. It can be observed that the CBQ structure shows the lowest sensitivity among the three presented configurations when $k = 10$. More precisely, the sensitivity on the central frequency has been reduced by 38 % in comparison with the non-optimized CBQ structure.

From Fig. 4, it is obvious that the increase in k results in lower sensitivity, but introduces previously mentioned drawbacks. Although sensitivity can be analytically observed and quantified, the effects arising from a higher ratio of component values represent a complex function of multiple variables. Therefore, a detailed investigation of these effects is beyond the scope of this paper and is left for future work.

However, it can be seen that the non-optimized CBQ structure lies on the steep part of the curve ($k = 1$). This means that the slight variations in k yield relatively rapid changes in total sensitivity of the filter. Also, as the k approaches the edge of the defined domain, the curve significantly flattens which means that with the further increase of k, the reduction in sensitivity becomes negligible.

In order to quantify this fact, Fig. 6 shows the derivative of the multiparameter sensitivity presented in Fig. 4. Since the derivative is representing a rate of change, it can be determined a much better tradeoff for practical implementation of the analyzed filter.

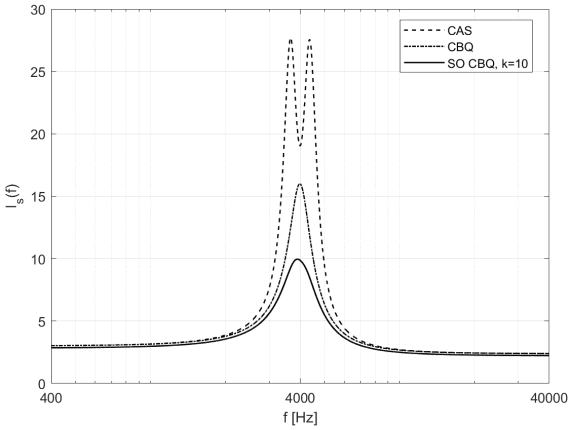


Fig. 5. Comparison of sensitivities for CAS, CBQ, and optimized CBQ.

As shown in Fig. 6, the local extrema of the multiparameter sensitivity occur at the points where the corresponding derivative equals zero. The derivative curve therefore quantifies the rate of change of the multiparameter sensitivity with respect to variations in k. It can be observed that selecting $k = 5$ results in only a marginal increase in sensitivity compared to $k = 10$, while simultaneously reducing the resistor and capacitor ratios by a factor of two.

The element values were once again recalculated, and the results are shown in Table 1. The sensitivity curve for this case is shown in Fig. 7, where is shown that there is indeed a slight increase in sensitivity, but with the benefit of half the ratio of components. The increase in sensitivity where $k = 5$ compared to where $k = 10$ is 6 %, while the decrease in comparison with the non-optimized CBQ structure is 34 %. The reason for choosing exactly k = 5 is on purpose to achieve better element ratios remaining low sensitivity. Non-integer values of k wouldn't result in any standard capacitor values, while this case gives easily implementable solution.

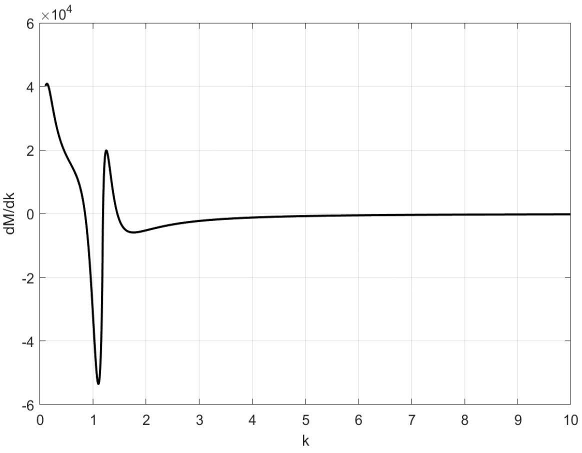


Fig. 6. Derivative of the multiparameter sensitivity.

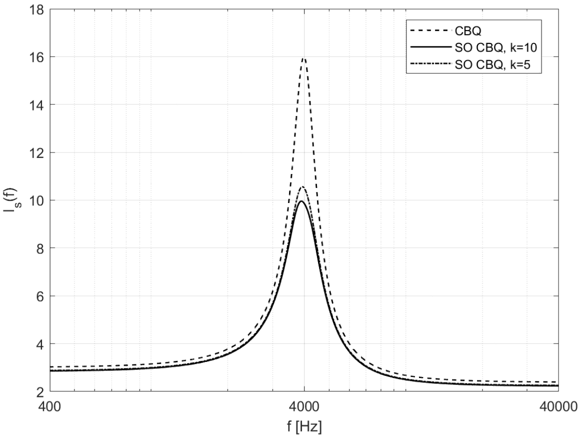


Fig. 7. Comparison of sensitivities for CBQ and optimized CBQ structures.

4. Simulation

As a validation and confirmation of the results obtained in MATLAB, the simulation is conducted in LTspice. Monte Carlo analysis is used with 100 passes and 1 % tolerance in element values.

The analysis is done for CAS, CBQ, and optimized CBQ (k = 10) cases. The results are shown in Fig. 8, Fig. 9 and Fig.10, respectively. As anticipated, the optimized CBQ structure exhibits the narrowest spread of frequency responses, thereby confirming the effectiveness of the applied optimization procedure.

Finally, Fig. 11 shows the circuit diagram of the CBQ bandpass filter structure analyzed and optimized in this paper.

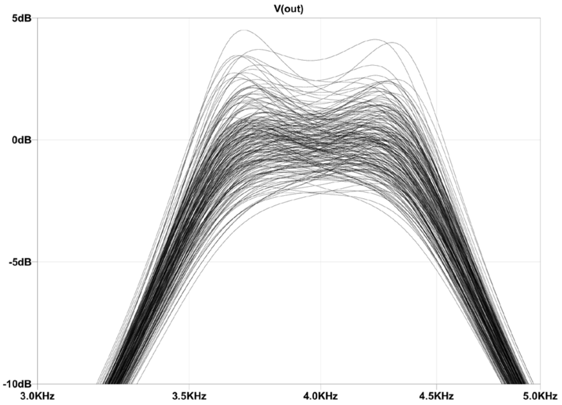


Fig. 8. Monte Carlo analysis of the CAS structure.

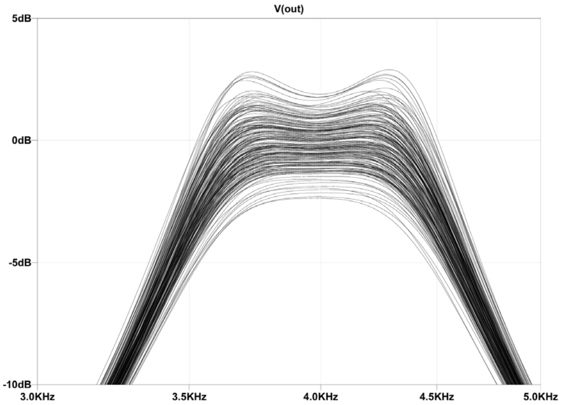


Fig. 9. Monte Carlo analysis of the CBQ structure.

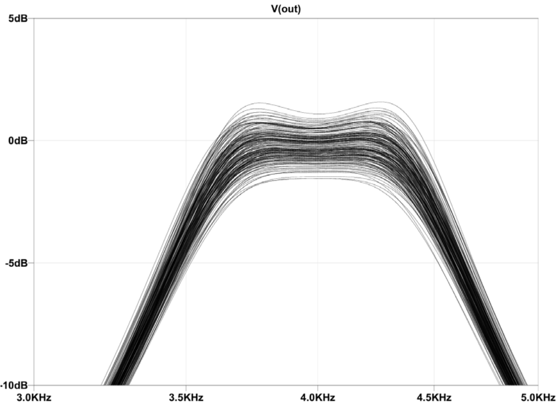


Fig. 10. Monte Carlo analysis of the optimized CBQ structure, $k = 10$.

5. Conclusions

In this paper, a fourth-order bandpass filter is analyzed and its sensitivity is optimized. It is shown that the CBQ structure inherently brings lower sensitivity at the expense of computational complexity and higher cost. Two optimization cases were done: one optimal on the defined domain of ratio k and the other as a compromise between minimal sensitivity and preserving lower ratio k. The optimization is conducted in MATLAB and verified in LTspice with Monte Carlo simulation. Further work may include quantifying the 'cost' of component ratio, and investigation of more complex filter topologies as well as optimization of the noise figure.

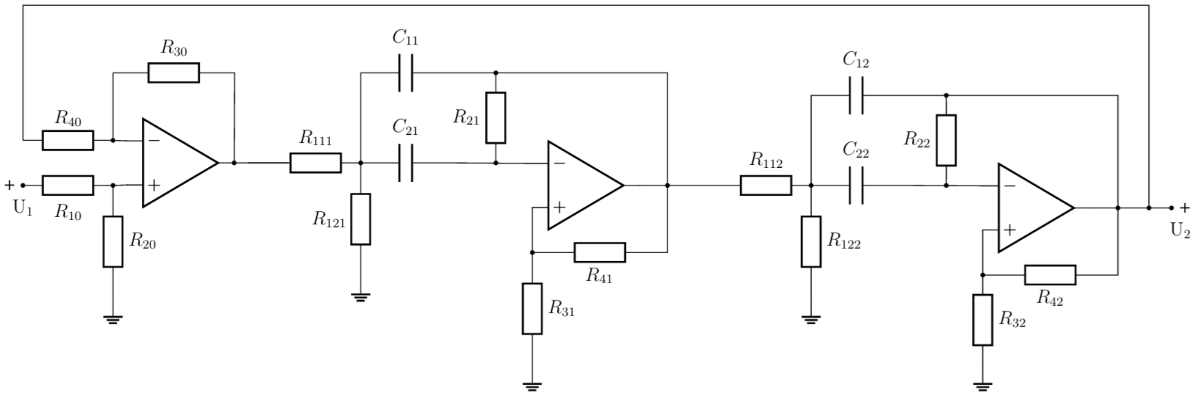


Fig. 11. Circuit diagram of the CBQ bandpass filter with a differential amplifier.

Acknowledgements

Funded by the European Union – NextGenerationEU – SDAFEM – uniri-mz-25-30.

References

[1]. Y. Fan, L. Zhao, J. Qin, J. Li, et al., High-precision time measurement electronics using the bandpass sampling method, *Review of Scientific Instruments*, Vol. 95, Issue 1, 2024, 014704.

[2]. S. Cecchi, V. Bruschi, S. Nobili, A. Terenzi, et al., Crossover networks: A review, *Journal of the Audio Engineering Society*, Vol. 71, Issue 9, 2023, pp. 526-551.

[3]. Z. A. Emon, K. M. Salim, I. B. Chowdhury, Design and analysis of a high-gain, low-noise, and low-power analog front end for electrocardiogram acquisition in 45 nm technology using g_m/I_D method, *Electronics*, Vol. 13, Issue 11, 2024, 2190.

[4]. J. Zhang, S. C. Chan, H. Li, N. Zhang, et al., An area-efficient and highly linear reconfigurable continuous-time filter for biomedical sensor applications, *Sensors*, Vol. 20, Issue 7, 2020, 2065.

[5]. N. Stojković, E. Kamenar, M. Šverko, Optimized second- and fourth-order LP and BP filters, *Automatika*, Vol. 52, Issue 2, 2011, pp. 158-168.

[6]. Z. Šverko, N. Stojković, S. Vlahinić, I. Markovinović, Noise and sensitivity comparison for different BP filter designs, *Automatika*, Vol. 62, Issue 3-4, 2021, pp. 319-330.

[7]. E. Živić, N. Stojković, S. Sladić, Sensitivity optimization method for 4th order CBQ structure LP filter, *Engineering Review*, Vol. 42, Issue 2, 2022, pp. 79-90.

[8]. J. D. Schoeffler, The synthesis of minimum sensitivity networks, *IEEE Transactions on Circuit Theory*, Vol. 11, Issue 2, 1964, pp. 271-276.

(005)

Graph Convolutional Networks in Recognition of Persuasive Faces in Online Media

K. Machová, M. Mach and P. Demeter
Technical University of Košice, Institute of Artificial Intelligence, Letná 9, 04200 Košice, Slovakia
Tel.: + 421 55 6024142
E-mail: kristina.machova@tuke.sk

Summary: This article focuses on the design, training, testing, and subsequent implementation of models of deep graph convolutional neural networks suitable for recognizing convincing faces from image data. The motivation for this work is its potential future use in detecting overly motivated authors of certain posts on social networks in the form of image data obtained from photographs or videos. This recognition of persuasive faces could be used in the prevention of fraud in the online space, where too much emphasis is placed on trying to convince people about a product or in a political campaign. The best results of graph convolutional neural networks were achieved in the case of the DLmodel12 model with a configuration of GCNConv layers 64, 32, 16, 8, and 4, where a recall of 92.68 % was achieved. Recall was a key metric, as our goal was to minimize False Negative (FN) error. However, other models also achieved relatively good results, especially in the positive class – persuasive training examples.

Keywords: Deep learning, Graph convolutional networks, Visual data processing, Persuasion classification, Online media analysis.

1. Introduction

Nowadays, we are witnessing an intensification of the spread of unhealthy phenomena such as conspiracy theories, disinformation, and opinion manipulation in the online space. This is often associated with videos where someone tries to convince us of something they may not even believe in themselves. An attentive listener can tell from body language and facial expressions that something is not right. We have focused on creating detection models that can recognize excessive persuasion, whether for the purchase of expensive products or in the promotion of conspiracy theories. In this way, we want to contribute to the user's ability to be critical and cautious.

Recognizing persuasive and neutral faces is a fascinating but challenging problem that could bring transparency and help reduce fraud. The ability to correctly interpret a neutral expression and distinguish it from persuasive facial expressions can have a wide range of applications, from improving human-computer interaction to enhancing security systems.

In recent years, innovative technology has begun to be used in neural networks and deep learning: graph neural networks, which offer new possibilities for more efficient and accurate recognition and interpretation of data in the form of graphs.

We focused on how graph neural networks can be applied to facial expression analysis. In general, we can say that, from a theoretical point of view, graph neural networks are particularly well suited for this purpose, as facial expressions can be naturally represented as graphs, where nodes represent important points on the face (e.g., corners of the mouth, ends of the eyebrows, etc.) and edges represent relationships or distances between these points. This ability to capture complex relationships between different parts of the face gives graph neural networks the ability to successfully recognize differences in the expressions of a convincing and neutral face.

2. State of the Art

2.1. Related Works

Graph neural networks (GNNs) are a class of neural networks designed to process data represented as graphs. They use the graph structure to aggregate information from neighboring nodes and edges. There are various methods such as Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and Message Passing Neural Networks (MPNN) [1]. The GNNs are used for processing relational data in various domains, for example social networks, face recognition, text processing, computer vision, or molecular structures, where analysis is focused on not only what the individual parts are, but also how they are connected to each other. They leverage graph structure to aggregate information from neighboring nodes and edges for tasks such as node property prediction, graph classification, or graph generation.

Graph Convolutional Network (GCN) is a neural network class suitable for image data processing. This type of GCN network has already been used in the field of image processing, but only in the medical field [2] and in the field of geodetic and spatial data [3, 4]. GCN networks have also been used to solve problems like the one we want to solve, namely the transformation of features extracted by the SVM model for converting neutral faces into convincing ones.

Peng et al. [5] introduced a multi-view learning framework based on graph convolution networks that integrates diverse graph representations and deep feature fusion techniques. Their model demonstrates that combining multiple feature views can significantly improve classification performance in semi-supervised learning scenarios. Such multi-view strategies are particularly relevant in facial recognition tasks where different feature types may be extracted from image data.

Bi and Dornaika [6] investigated a graph-based semi-supervised learning approach that integrates linear projection with multi-view data analysis. Their work focuses on improving classification performance in situations where only a limited amount of labelled training data is available. This problem is common in many facial recognition datasets and therefore represents an important methodological contribution.

Sun et al. [7] presented a method for signal classification based on variational Bayesian learning combined with graph convolutional networks. Although their work focuses on electroencephalogram signal analysis, the proposed integration of probabilistic learning with graph convolution demonstrates advanced optimization strategies that may also benefit other graph-based recognition tasks.

In the context of face recognition, study [8] focused on recognizing faces with masks on. This research used the research methodology of artifact (model) design and development, i.e., modeling and experimentation. As for the model itself, a pre-trained deep convolutional neural network model ResNet was used. However, instead of the output layer, three machine learning models were used on the final layer of the model, namely SVM, decision tree, and a set of classifiers. In terms of the performance of individual models, all achieved an accuracy of over 90 %. However, in terms of the ratio of accuracy and time to train the models, the best results were achieved by a combination of a neural network with an SVM classifier at the output.

Another article that inspired us [9] focused on generating a certain type of face. Specifically, this research concerned the design and development of an artifact, namely a convincing virtual agent generated from neutral faces. This artifact was named THRUST. The goal of this work was to create a convincing virtual agent that would inspire older people to engage in physical activity, thereby contributing to the improvement of their health. This tool also includes a classification model for classifying neutral and convincing faces using machine learning. The SVM method was used to detect neutral and persuasive faces. The approach was based on modeling the nonverbal features of a persuasive face. A convolutional model was used to transform features from a neutral to a persuasive face.

A similar problem was addressed in [10]. The authors proposed a conditional variational autoencoder to generate visually distinct faces by transforming neutral faces into new ones that were more appropriate for use in advertising.

What is new about our research is that we do not aim to generate convincing faces, but rather to detect them among ordinary, neutral faces. We want to design a GCN-based model capable of identifying generated synthetic convincing faces or modified faces that may be created with malicious intent. We want to offer a way to mitigate this negative phenomenon.

2.2. Graph Convolutional Neural Networks

Graph Convolutional Networks (GCNs) is a neural network class suitable for image data processing. There are two types of GCNs - spectral and spatial. GCNs are a special type of convolutional neural networks that do not work with input data in the form of vectors or matrix but in the form of graphs. GCNs are referred to as the transformation of all graph elements, i.e., nodes, edges, and relationships between these objects. Convolution in the case of graph data is applied in the form of a graph Fourier transformation. This is defined as the projection of the values of the Laplace operator. As in the case of classic convolutional neural networks, graph convolutional neural networks are generally built from convolutional and pooling layers for feature extraction, which are followed by fully connected layers. Subsequently, for feature classification, we use classic SoftMax and sigmoid classification output layers [11]. The difference between classic convolutional neural networks (CNN) and graph convolutional neural networks GCN is illustrated in Fig. 1.

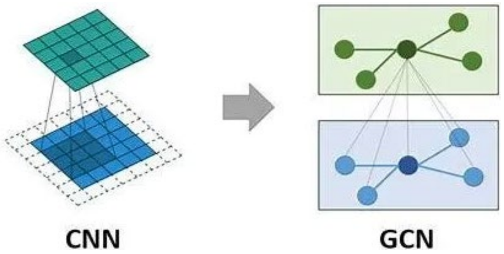


Fig. 1. Illustration of difference between CNN and GCN.

The basic building block of GCNs is the Graph Convolution Layer (GCNConv), which represents an advanced approach to learning on graph structures. This concept is based on the idea that information from nodes in a graph can be effectively propagated and aggregated through edges, allowing the model to learn complex patterns dependent on the structure and properties of graphs. GCN networks focus on processing graph data, where the data contains nodes and edges. Nodes represent entities and edges represent relationships between them. GCNs apply deep learning concepts to such structured data, allowing models to learn complex patterns in the relationships between entities. The core of GCNs lies in "message propagation," a process where nodes communicate with their neighbors. Each node receives information (messages) from its neighbors and processes it using aggregation, such as sum, average, or maximum value. Subsequently, the state of the node is updated.

This process is repeated iteratively and can be performed on multiple layers in a deep neural network. Learning GCNs involves optimizing model parameters and weights of individual neurons so that the model effectively maps input graphs to obtain outputs, such as node or graph labels [11].

The basic principles of the GCNConv layer include:

- *Aggregation*: Aggregation functions (such as sum, mean, or max) are used to work with data. We use these aggregation functions to merge information from multiple neighboring nodes into a single vector. In this way, GCNConv captures and simplifies information from the local neighborhood of a node, which serves to make more efficient use of computing power and thus simplify models;
- *Node state update*: After aggregating neighboring information, the new node state is calculated by combining its original state and the aggregated information, often through a linear transformation followed by a nonlinear activation function;
- *Recursive application*: The aggregation and update process can be applied multiple times (i.e., across multiple layers of GCNconv), allowing the model to obtain information from progressively larger neighborhoods of nodes;

The output H(l+1) of the GCNConv graph convolutional layer can be calculated from previous state H(l) using equation (1).

$$\mathrm{H(l+1)} = \sim D^{-\frac{1}{2}} * \sim A * \sim D^{-\frac{1}{2}} * H(l) * * W(l) + b, \tag{1}$$

where *H(l)* is the input feature matrix for layer "*l*", $\sim A$ is adjacent matrix with added self-loops, $\sim D$ is diagonal degree matrix of $\sim A$, *W(l)* is the weight matrix for layer "*l*", and *b* is the bias term for layer "*l*".

The principle of GCN is therefore based on searching for node properties and relationships between them, to which aggregation functions are then recursively applied. The principle of GCN networks is illustrated in Fig. 2.

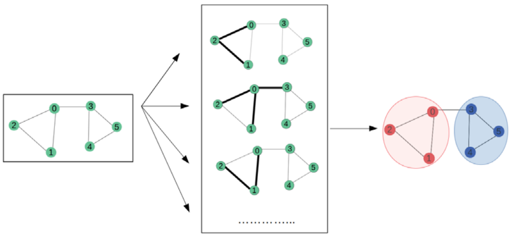


Fig. 2. Illustration of the principle of GCN [12].

Our focus is on applying GCN in the field of computer vision, i.e., image data processing. Study [1] presents the possibility of using GCN in image data classification, where, after applying appropriate construction models, it is possible to obtain structured graphs from unstructured images, which we can use for GCN training. The second possibility of using GCN on image data is to obtain image descriptions, or to train a neural network model to answer questions about the inserted image. Another way of using GCN is one in which it is possible to capture relationships between objects in an image.

3. Methodology

The methodology of our research is focused on suitable data acquisition and pre-processing. Then the data were used for deep GCN training using the GCNConv layer from the library Spectral as essential part of the model for persuasive faces recognition.

3.1. Data Description

We created the dataset for training GCN by extracting data from two datasets. The first one was the dataset that is freely available on kaggle.com [13]. This is a dataset of faces representing various emotions: disgust, fear, happiness, sadness, surprise, as well as emotionally neutral faces. From these images of faces, we created two classes: a neutral class, where we included a group of images labeled as neutral faces. We created the second class of persuasive faces by selecting suitable faces from the other emotions.

The second dataset we used contained images of the faces of US senators and congressmen. After visually checking the images of these politicians' faces, we classified them as belonging to the persuasive class. This dataset is freely available at tlab.uchicago.edu/ db-redirect-politicians-faces/ [14]. The dataset created in this way contained 6198 images of neutral faces, and there were 9535 images in the class of persuasive faces.

We then focused our data analysis on the size of individual images and their color palette. In terms of image dimensions, we looked for the largest, smallest, and average dimensions so that we could then resize the images to the same size. The smallest image height and width were 48 px, meaning that the smallest images had dimensions of 48×48 px. As for the dimensions of the largest images, they had a width of 165 px and a height of 212 px. The average image width was 150 px and the average image height was 207.86 px. As for the color palette of the images, we found that the images were stored in two color formats: RGB images (color) and black-and-white images.

We decided to convert all images to black and white format, as most of the images were already in black and white format. The second issue was the varying size of the images. Since the average image size was closer to the maximum dimensions, we decided to resize the images to 255×255 px.

The next step in preprocessing the data for graph convolutional networks was to create facial feature graphs. We did this using the *face_recognition library*. After loading the data, we used the *face_recognition.face_landmarks* function for each image to automatically capture facial features along with the emotion expressed on the face, i.e., a convincing or neutral class. An example of the facial features created can be seen in Fig. 3.

Fig. 3. Illustration of the creation of facial features.

The next step was to extract graphs from facial features and convert these graphs into numerical formats. We performed the graph extraction again using the *face_recognition.face_landmarks* function and then converted them into numerical format using the *toarray()* function from the same library. The extracted graphs were in a format where they contained 9 nodes (left_eye, nose_bridge, right_eye, left_eyebrow, right_eyebrow, nose_tip, top_lip, bottom_lip, chin), and each of these nodes were described by 9 properties. The last two steps in the preprocessing of data for GCN were data normalization and the creation of an adjacency matrix. We normalized the data using the *StandardScaler* function from the *sklearn.preprocessing library*. In Fig. 4, we can see an example of a graph created from extracted facial features.

We then converted this data into numerical format, i.e., *numpy* format. This data was used to train GCN networks. An example of one training example can be seen in Fig. 5.

3.2. GCN Network Training

Before starting GCN training, the data was divided into training, testing, and validation sets. Validation data are necessary for evaluating models during the training phase. Validation data allows to monitor the model's feedback on an unknown data set during training. Based on the evaluation of data in the validation set, it is possible to tune neural network models by adjusting individual layers of the model as well as its hyperparameters. The test data set is used to test the model's results after the training and hyperparameter tuning phase is complete. This is data that the model has never seen before, so it provides the opportunity to measure the results and success of the trained models without bias. The data was divided into a ratio of 70/10/20.

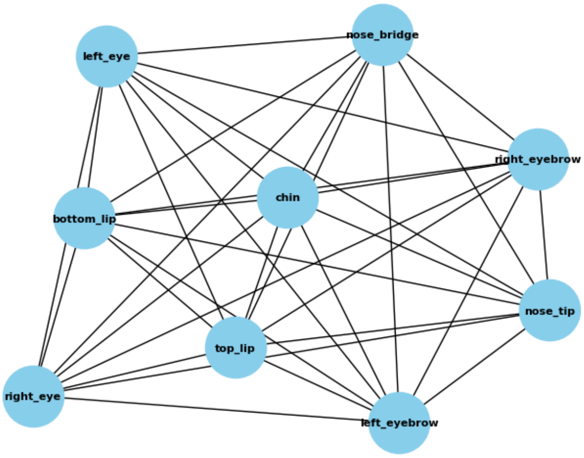
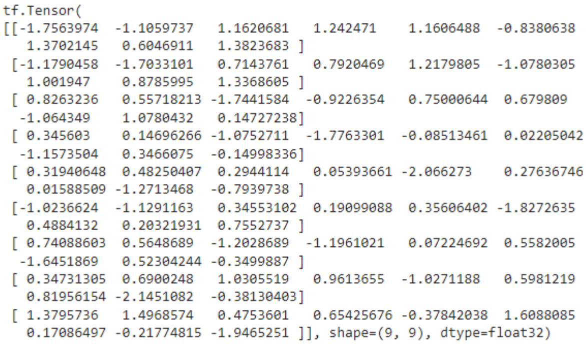


Fig. 4. The graph was created from extracted facial features.



```
tf.Tensor(
[[-1.7563974  -1.1059737   1.1620681   1.242471    1.1606488  -0.8380638
   1.3702145   0.6046911   1.3823683 ]
 [-1.1790458  -1.7033101   0.7143761   0.7920469   1.2179805  -1.0780305
   1.001947    0.8785995   1.3368605 ]
 [ 0.8263236   0.55718213 -1.7441584  -0.9226354   0.75000644  0.679809
  -1.064349    1.0780432   0.14727238]
 [ 0.345603    0.14696266 -1.0752711  -1.7763301  -0.08513461  0.02205042
  -1.1573504   0.3466075  -0.14998336]
 [ 0.31940648  0.48250407  0.2944114   0.05393661 -2.066273    0.27636746
   0.01588509 -1.2713468  -0.7939738 ]
 [-1.0236624  -1.1291163   0.34553102  0.19099088  0.35606402 -1.8272635
   0.4884132   0.20321931  0.7552737 ]
 [ 0.74088603  0.5648689  -1.2028689  -1.1961021   0.07224692  0.5582005
  -1.6451869   0.52304244 -0.3499887 ]
 [ 0.34731305  0.6900248   1.0305519   0.9613655  -1.0271188   0.5981219
   0.81956154 -2.1451082  -0.38130403]
 [ 1.3795736   1.4968574   0.4753601   0.65425676 -0.37842038  1.6088085
   0.17086497 -0.21774815 -1.9465251 ]], shape=(9, 9), dtype=float32)
```

Fig. 5. An example of one training example of data for GCN network training.

As part of model training, we focused on optimizing parameters, using different settings for optimization and regularization, and using different settings and numbers of individual layers.

First, we tried to find out which combination of *GCNConv* graph convolutional layers from the *Spectral library* would give the best results for the trained neural network. In all experiments, we used binary cross-entropy as the loss function, which is ideal for binary classification. Initially, we experimented with combinations of two hidden graph convolutional neural networks with the following graph convolutional layer hyperparameter settings: 16, 32, 64, 128, 256, and 512. Among these values, we tested all different combinations of settings for the number of neurons within two hidden layers with the same setting. We started these experiments with a number of epochs 50 and a batch size of 32. Based on these experiments, we selected the best combinations of GCN, on which we then perform experiments with different settings of optimization and regularization techniques and functions. We extended the experiments to different learning rate settings for the Adam optimization function and to test the deployment of SGD optimization. When using SGD, the experiments were unsuccessful. In general, we can say that the best results were achieved by setting the learning rate hyperparameter to 0.01.

Table 1 shows the architecture of the six best models, including the settings of the hyperparameters of the graph convolutional layers. These graph convolutional layers are followed by a GAP – *GolbalAvaragePooling1D* layer and a *Sigmoid* output layer for binary classification.

Table 1. The architecture of the six best models.

Model	GCNConv1	GCNConv2	Output
7	32	64	GAP 1D+Sigmoid
12	64	256	GAP 1D+Sigmoid
22	128	64	GAP 1D+Sigmoid
20	256	128	GAP 1D+Sigmoid
17	256	512	GAP 1D+Sigmoid
19	512	256	GAP 1D+Sigmoid

4. Results of Experiments

We evaluated the results of experiments with the six models listed in Table 1. As part of this, we verified which batch size parameter setting would work best when training GCN. The results are shown in Table 2 and Table 3, where B indicates the batch size parameter setting value, Acc is short for Accuracy, and F1 represents the F1-score.

Table 2. The results of first three of six best models in Acc and F1 when changing batch size (B) from 16 to 256.

		Model 7	Model 12	Model 22
B = 16	Acc	0.8453	0.8264	0.8480
	F1	0.8350	0.8051	0.8379
B = 32	Acc	0.8492	0.8476	0.8472
	F1	0.8390	0.8348	0.8342
B = 64	Acc	0.8433	0.8468	0.8421
	F1	0.8306	0.8354	0.8348
B = 128	Acc	0.8260	0.8445	0.8476
	F1	0.8214	0.8354	0.8363
B = 256	Acc	0.8390	0.8464	0.8409
	F1	0.8313	0.8350	0.8322

The results in Table 2 and Table 3 show that although the results were close, when considering the results of all models in experiments with the batch size parameter, i.e., the data sample in 1 training epoch, the best setting for the batch size parameter was 128.

Table 3. The results of second three of six best models in Acc and F1 when changing batch size (B) from 16 to 256.

		Model 20	Model 17	Model 19
B = 16	Acc	0.8445	0.8464	0.8449
	F1	0.8378	0.8341	0.8312
B = 32	Acc	0.8468	0.8492	0.8495
	F1	0.8361	0.8397	0.8361
B = 64	Acc	0.8433	0.8421	0.8476
	F1	0.8349	0.8402	0.8402
B = 128	Acc	0.8433	0.8457	0.8457
	F1	0.8342	0.8380	0.8342
B = 256	Acc	0.8425	0.8496	0.8468
	F1	0.8346	0.8389	0.8368

Further experiments we conducted concerned setting the learning rate parameter for the Adam optimization function. As part of these learning rate parameter settings, we also tested the use of a learning rate scheduler. For these experiments, we used only the top 6 models: 7, 12, 22, 20, 17, and 19.

Based on the results of these experiments, we can conclude that the best results were achieved when the learning rate was set to 0.01, or even when using a learning rate scheduler. When the learning rate was set to 0.1 and 0.05, some models experienced oscillations and large fluctuations during neural network training. This is shown in Fig. 6 (learning curve at a learning rate of 0.01) and Fig. 7 (learning curve at a learning rate of 0.1).

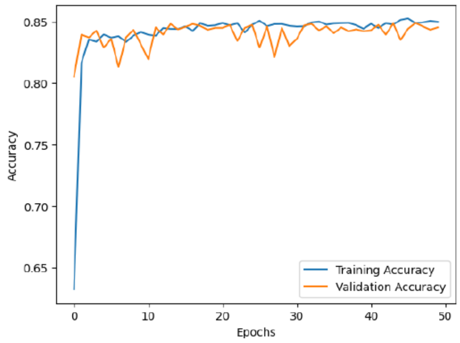


Fig. 6. Illustration of learning curve at a learning rate of 0.01 for GCN network training.

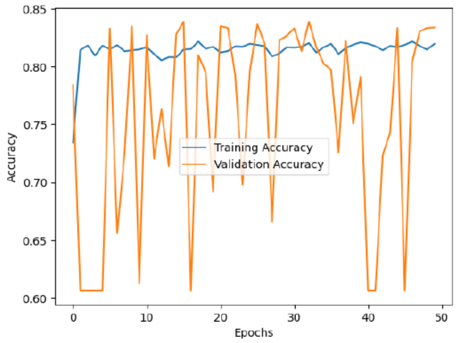


Fig. 7. Illustration of learning curve at a learning rate of 0.1 for GCN network training.

5. Conclusions

The aim of this research is to provide an overview of the possibilities of using graph convolutional neural networks in image data processing and to focus their use on solving the problem of recognizing persuasive and neutral faces in social media. The article presents the results of experiments with various configurations and architectures of graph convolutional neural networks in the detection of persuasive faces from image material.

The GCN experiments were based on the use of the GCNConv graph convolutional layer from the Spectral library. As part of these experiments, we tested various settings for hidden layers, error, optimization, and regularization functions. We then evaluated them. Model tests showed that we achieved results comparable to results of other researchers on the same benchmark when focusing on the issue of image classification using GCNs.

The best GCN models have the potential to be deployed as a supporting or auxiliary tool in detecting persuasive faces for the purpose of regulating the number of frauds or manipulative content on social networks.

Acknowledgements

This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-22-0414 MODERMED “Multimodal Detection of Toxic Behaviour in Social Media”.

References

[1]. S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: A comprehensive review, *Computational Social Networks*, Vol. 6, Issue 1, 2019, pp. 1-11.

[2]. G. Zhang, J. Pan, Z. Zhang, H. Zhang, et al., Hybrid graph convolutional network for semi-supervised retinal image classification, *IEEE Access*, Vol. 9, 2021, pp. 35778-35789.

[3]. L. Mou, X. Lu, X. Li, X. X. Zhu, Nonlocal graph convolutional networks for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 58, Issue 12, 2020, pp. 8246-8257.

[4]. A. Abbas, M. M. Abdelsamea, M. Gaber, Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network, *Applied Intelligence*, Vol. 51, Issue 2, 2021, pp. 854-864.

[5]. G. Peng, F. Dornaika, J. Charafeddine, Multi-view learning with graph convolution networks adopting diverse graphs and genuine deep feature fusion, *Artificial Intelligence Review*, Vol. 58, 2025, 290.

[6]. J. Bi, F. Dornaika, Linear projection fused graph-based semi-supervised learning on multi-view data, *Artificial Intelligence Review*, Vol. 58, 2025, 309.

[7]. X. Sun, Y. Li, X. Zhai, et al., Accurate electroencephalogram classification by variational Bayesian learning and graph convolutional networks, *EURASIP Journal on Advances in Signal Processing*, Vol. 2026, 2026, 9.

[8]. M. Loey, G. Manogaran, M. H. N. Taha, N. E. M. Khalifa, A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic, *Measurement*, Vol. 167, 2021, 108288.

[9]. A. Cherni, R. Bertrand, M. Ochs, From neutral human face to persuasive virtual face, a new automatic tool to generate a persuasive attitude, in *Proceedings of the 4th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI)*, 2022, pp. 58-61.

[10]. C. Thomas, A. Kovashka, Persuasive faces: Generating faces in advertisements, *arXiv*, 2018, arXiv:1807.09882.

[11]. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[12]. I. Mayachita, Understanding graph convolutional networks for node classification, Medium, https://medium.com/data-science/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b

[13]. V. Shayal, S. Mukund, Face emotions data, Kaggle, 2022, https://www.kaggle.com/datasets/shayalvaghasiya/face-emotions-data

[14]. C. Y. Olivola, A. Todorov, Elected in 100 milliseconds: Appearance based trait inferences and voting, *Journal of Nonverbal Behavior*, Vol. 34, 2010, pp. 83-110.

(007)

When Does a Kalman Filter Beat an LSTM?

Arthur Debauge [1], **Noria Foukia** [1] **and Clément Denis** [2]
[1] HESSO – HEPIA – ISC 4, rue de la Prairie, 1202, Geneva, Switzerland
[2] HESSO – HEPIA – GC 4, rue de la Prairie, 1202, Geneva, Switzerland
E-mail: arthur.debauge@hesge.ch, noria.foukia@hesge.ch, clement.denis@hesge.ch

---

Summary: We benchmark Cubic Splines, Kalman Filter, LSTM, and Kalman + LSTM on the UCI Household Power Consumption dataset. The Kalman filter achieves R^2 = 0.957, $\|E\|_F$ = 6.95, outperforming all LSTM variants and a naive persistence baseline. Kalman pre-processing reduces LSTM's Frobenius error by 13.4 % at no architectural cost. SHAP confirms the Kalman-filtered signal as the dominant predictor (54.9 %), justifying the hybrid architecture. We characterise when this holds and provide a principled comparison framework using the Frobenius norm as optimisation criterion.

Keywords: Energy forecasting, Kalman filter, LSTM, Frobenius norm, SHAP, Microgrid.

---

1. Introduction

Short-term energy load forecasting underpins P2P trading decisions in smart microgrids. The literature favours LSTM-based architectures [2, 7], yet their advantage over classical estimators is rarely quantified on real household data [6]. Our central finding is counter- intuitive: a well-tuned Kalman filter beats all LSTM variants on the Frobenius norm, because the underlying signal is quasi-Markovian ($\rho_{t,t+1}\approx 0.97$). We characterise when this holds and show that, when LSTM is warranted, Kalman pre-processing is an improvement at no architectural cost.

2. Dataset & Setup (Fig. 1)

UCI Household Electric Power Consumption [1] (France, 2007–2010, 1-min → 2 h), enriched with ERA5 meteorological covariates. Regression target: *Global_active_power(t+1)* in kW. Features: 7 electrical + 4 meteorological + 4 cyclic time encodings. Temporal split: 70/15/15 (train/val/test), strictly causal.

3. Methods (Fig. 2)

3.1. Persistence Baseline

Naïve baseline: $\hat{y}_{t+1} = y_t$. With $\rho_{t,t+1}$ = 0.972 on the test set, this is a strong lower bound against which all methods are compared. Any model failing to outperform persistence provides no practical value.

3.2. Kalman Filter (Order 2)

Constant-velocity state $x_k = [pos., vel.]^T$:

$$\hat{x}_k = \hat{x}_k^- + K_k\,(z_k - H.\hat{x}_k^-),$$
$$K_k = \frac{P_k^-.H^T}{H.P_k^T.H^T+R}$$

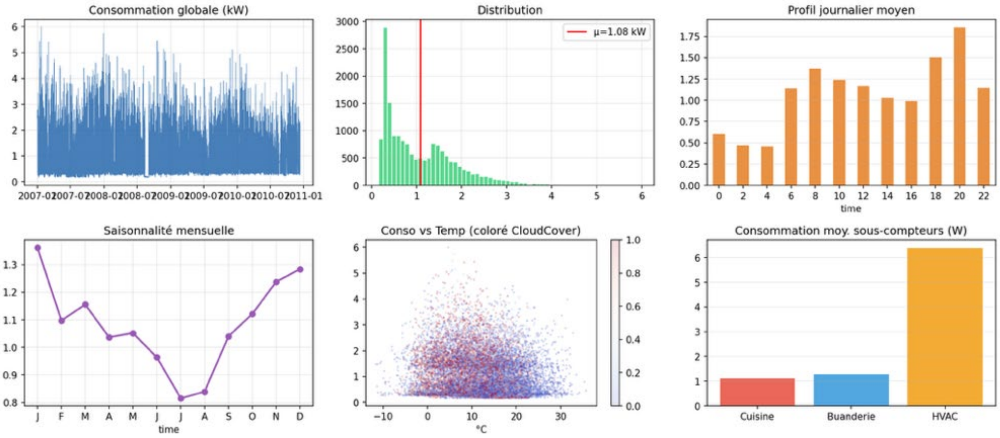


Fig. 1. Exploratory Data Analysis of the electrical consumption dataset. Highlights include the global time series (2007–2011), daily load profiles, monthly seasonality, and the distribution of sub-metering, with HVAC showing the highest energy demand.

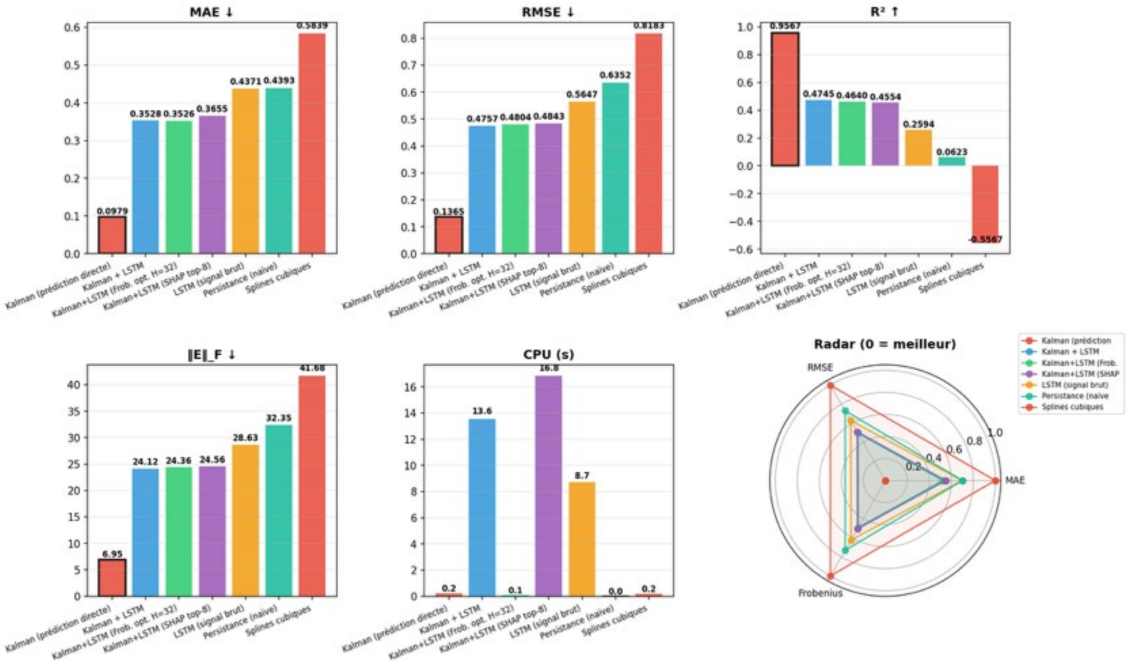


Fig. 2. Comparative benchmark of prediction models across multiple metrics (MAE, RMSE, R2, ‖E‖F, and CPU time). The Kalman filter (direct prediction) outperforms LSTM and Spline-based approaches in both accuracy (R2 = 0.9567) and computational efficiency.

3.3. LSTM

Single-layer LSTM [2]: T = 24 steps, Adam (3.10^{-3}, batch 64), early stopping (patience 6).

3.4. Kalman + LSTM

Raw dataset replaced by its Kalman-filtered version. No architectural change, no added parameters.

3.5. Frobenius Optimisation

Grid search minimising $\|E\|_F = \sqrt{\sum_i e_i^2}$ on validation. $H \in \{16, 32, 64\}$, $l_r \in \{1.10^{-3}, 3.10^{-3}, 7.10^{-3}\}$. $\|E\|_F$ penalises large errors more heavily than MAE and measures a distance in the error space, making it a natural criterion for comparing models on a fixed test set.

4. Results & Discussion (Fig. 2 & Table 1)

Table 1 and Fig. 2 summarise all results.

Kalman beats LSTM. The Kalman predictorachieves $R^2 = 0.957$, $\|E\|_F = 6.95$ because $\rho_{t,t+1} \approx 0.97$: a quasi-Markovian signal has no systematic information beyond what a Bayesian recursive estimator already captures [3, 5]. *Rule:* benchmark Kalman before LSTM whenever $\rho_{t,t+1} > 0.9$. LSTM adds value for multi-step horizons or heterogeneous feature sets.

Persistence baseline. The naive persistence predictor yields $\|E\|_F = 32.35$, $R^2 = 0.062$, confirming that the high autocorrelation alone does not trivially solve the forecasting problem. The Kalman filter outperforms persistence by a factor of ≈4.6 in $\|E\|_F$, while the raw LSTM offers only a modest improvement ($\|E\|_F = 28.63$).

Table 1. Test-set benchmark sorted by ‖*E*‖*F* (ascending), **Bold** = best per metric, Green = dominant method, Blue = persistence baseline (added per reviewer request). ‖*E*‖*F* = √(∑*i* *ei*²). CPU measured on a single core.

Method	MAE (kW)	RMSE (kW)	R^2	$\|E\|_F$	CPU (s)
Kalman filter (direct)	**0.0979**	**0.1365**	**0.9567**	**6.952**	0.2
Kalman + LSTM	0.3497	0.4780	0.4694	24.237	19.6
Kalman+LSTM (SHAP top-8)	0.3574	0.4787	0.4678	24.273	10.1
Kalman+LSTM (Frob. opt.)	0.3591	0.4833	0.4577	24.503	0.1[a]
LSTM (raw signal)	0.4227	0.5521	0.2921	27.996	10.1
Persistence ($\hat{y}_{t+1}=y_t$)[b]	0.439	0.635	0.062	32.35	<0.1
Cubic Splines	0.5839	0.8183	−0.5567	41.684	0.2

[a] Inference latency only (best model already selected by grid search); training cost of the 3×3 grid search is excluded from this value.

[b] Added as lower-bound baseline; lag-1 autocorrelation on the test set: $\rho_{t,t+1}$=0.972.

Kalman pre-processing improves **LSTM.** Replacing raw by filtered data reduces $\|E\|_F$: 28.0→24.2 (−13.4 %), R^2: 0.292→0.469 (+60.5 %), at no architectural cost [6].

SHAP. Filtered data holds 54.9 % of feature importance [4]. SHAP top-8 ($\|E\|_F$ = 24.27) equals Frobenius-optimised tuning ($\|E\|_F$ = 24.50) with 47 % fewer features.

Takeaway. Benchmark Kalman before LSTM ($\rho_{t,t+1}$ > 0.9); Kalman pre-processing is architecturally free; tune with $\|E\|_F$; audit with SHAP.

References

[1]. G. Hébrail, A. Bérard, Individual household electric power consumption data set, UCI Machine Learning Repository, 2012, https://archive.ics.uci.edu/dataset/235/individual-household-electric-power-consumption

[2]. W. Kong, et al., Short-term residential load forecasting based on LSTM recurrent neural network, *IEEE Transactions on Smart Grid*, Vol. 10, Issue 1, 2019, pp. 841-851.

[3]. R. E. Kalman, A new approach to linear filtering and prediction problems, *Journal of Basic Engineering*, Vol. 82, Issue 1, 1960, pp. 35-45.

[4]. S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in Advances in Neural Information Processing Systems 30 (I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, et al., Eds.), *Curran Associates, Inc*., Red Hook, 2017, pp. 4765-4774.

[5]. S. Sharma, A. Majumdar, V. Elvira, E. Chouzenoux, Blind Kalman filtering for short-term load forecasting, *IEEE Transactions on Power Systems*, Vol. 35, Issue 6, 2020, pp. 4916-4919.

[6]. T. Y. Kim, S. B. Cho, Predicting residential energy consumption using CNN-LSTM neural networks, *Energy*, Vol. 182, 2019, pp. 72-81.

[7]. Y. Wang, N. Zhang, X. Chen, A short-term residential load forecasting model based on LSTM recurrent neural network considering weather features, *Energies*, Vol. 14, Issue 10, 2021, 2737.

(008)

Rapid Adaptation of UXO Classification using Meta-Learning from Limited Side Scan Sonar Data

Franziska Auer [1], Ashik Shaji [1] and Tobias Meisen [2]
[1] PTI, TKMS ATLAS ELEKTRONIK GmbH, Sebaldsbruecker Heerstrasse 235,
28309 BREMEN, Germany
[2] TMDT, University of Wuppertal, Lise-Meitner-Straße 27-31, 42119 Wuppertal, Germany
Tel.: +49 421 457-4043
E-mail: Franziska.Auer@tkmsgroup.com

Summary: Automated detection and classification of objects in side scan sonar (SSS) images is essential for maritime safety, environmental monitoring, and unexploded ordnance (UXO) clearance, but progress is hampered by two intertwined challenges: The scarcity of accurately labeled sonar data and the strong variability of acoustic appearance across sensors, seabed conditions, and viewing geometries. This work investigates whether few-shot learning (FSL) is a viable and effective strategy for overcoming these challenges. To this end we assess several FSL methods (Deep Subspace Networks (DSN), ProtoNet, Matching Networks, and Relation Networks) within a meta-learning framework trained solely on sonar data, and compare backbones (Conv32f, Conv64f, ResNet12, ResNet18, and a YOLOv8m-based) using real-world sonar datasets and benchmarking against two conventional baselines (a fine-tuned YOLOv8m-cls classifier and a PCA-based classifier). Experiments reveal that features extracted with the YOLOv8m-cls backbone consistently outperform traditional Conv/ResNet backbones on both a meta-validation split and an out-of-distribution IRAV test set, confirming the benefit of a classification-specific backbone for sonar. ProtoNet combined with YOLOv8m-cls achieves the highest test performance (F1-score 83.89 %). A k-shot analysis shows that three support examples per class already surpass a 60 % accuracy threshold, demonstrating rapid adaptation with minimal labeling effort. The results substantiate that FSL is a promising, data-efficient approach for underwater object classification, while also highlighting the critical role of domain-tailored feature extraction and the need for diverse, sonar-specific training data to ensure robust cross-condition generalisation.

Keywords: Few shot learning, Sonar target detection, Unexploded ordnance.

---

1. Introduction

Classifying unexploded ordnance (UXO) correctly with only a few identified examples is a pressing challenge for maritime safety and cost-effective clearance. Conventional side scan sonar (SSS), a primary tool for seabed mapping, produces imagery degraded by variable seafloor conditions, sensor- and geometry-dependent acoustic shadows, and highly diverse object shapes and materials, which makes the creation of large, meticulously annotated datasets both time- and resource-intensive. Few-shot learning (FSL) offers a promising solution. By training a meta-learning model across many small tasks, it can generalize from a limited number of examples. In this work we realize that concept in a complete framework that is trained and evaluated exclusively on real-world sonar data (meta-training, meta-validation, meta-testing), ensuring that learned features are tuned to sonar imagery. Experiments demonstrate faster, more efficient underwater object classification than methods [1] that rely on RGB-image-based meta-training.

We conduct the first extensive FSL benchmark on exclusively real-world sonar data, introduce a classification specific YOLOv8m-cls backbone that consistently outperforms traditional Conv/ResNet encoders, and systematically compare four state-of-the-art FSL algorithms (Deep Subspace Networks (DSN), ProtoNet, MatchingNet, RelationNet). In addition, we provide baseline ablations against a fine-tuned YOLOv8m classifier and a principal component analysis (PCA) analysis, and a k-shot analysis showing that three labeled examples per class already surpass a 60 % accuracy threshold, thereby demonstrating the practical data efficiency of the approach.

The remainder of this paper is organized as follows: Section 2 reviews related work in underwater object detection using FSL and highlights the gaps in current research. Section 3 describes the dataset used. Section 4 details the experimental setup, including a description of the datasets used. Section 5 presents the results of our experiments ”Comparison FSL models”, ”Baseline Comparison”, and ”KShot Learning”. Finally, section 6 briefly summarizes the main findings of this paper and gives directions for future research.

2. Related Work

2.1. FSL Techniques

FSL enables models to recognize new categories from only a few labeled examples by leveraging prior knowledge from related tasks, thus mimicking human generalization. The prevailing paradigm is meta-learning, which trains across diverse tasks for rapid adaptation and is conventionally divided into metric-based, model-based, and optimisation-based methods.

Metric-based learning approaches, such as ProtoNet [2], MatchingNet [3], and RelationNet [4], focus on learning an embedding space and similarity metrics to compare few-shot examples for classification. ProtoNet [2] uses a metric-based method by learning an embedding space where classification is performed by calculating distances to class prototypes, which are mean embeddings of support samples. MatchingNet [3] extends this by employing an attention mechanism to compute cosine similarities between query and support examples. RelationNet [4] enhances this approach with a learnable relation module that outputs pairwise similarity scores, improving adaptability to new tasks.

Model-based learning methods, exemplified by DSN [5], embed prior knowledge about the data or domain structure within model architectures. DSN [5] incorporates domain-specific sub-networks that improve representation learning across varied tasks by modeling domain shifts explicitly.

2.2. FSL in Sonar and Related Domains

Limited annotated data and costly labeling are common in fields like medical imaging and remote sensing. The extensive research there offers robust frameworks, architectures, and meta-learning strategies that can be transferred to sonar. In a test performed on radiology medical images, model-agnostic meta-learning (MAML) [6] outperformed the other FSL techniques with an accuracy of 78.8 % during 3-way 3-shot metatesting [7]. Opposing to this, MAML [6] was outperformed by other FSL methods when Lu et al. used a breast tumor dataset (BreakHis [8]) and the ISIC 2018 skin lesion dataset [9]. During 3-way 3-shot meta-testing, RelationNet [4] obtained the best accuracy with 67.32 %, only being outperformed by the authors own model for whom the code is not available [10].

To overcome limitations within the sonar domain, Ochal et al. [1] tested different FSL models on synthetic SSS data. They achieved an accuracy of 61.4 % on the synthetic SSS test data with a soft k-means prototypical network. For meta-training a mixture of Mini-ImageNet [11] and the synthetic SSS data was used. Even though the results on SSS data stayed below the maximum accuracy they achieved on RGB images, they proved that using 5-shot 5-way FSL on SSS is a feasible option to distinguish well between new objects with only little data known of them.

Our research extends this success by substituting synthetic images with real sonar data gathered from multiple sensors, thereby supplying sufficient class diversity for meta-training and meta-validation and explicitly accounting for the risk that variability in acquisition conditions can produce markedly different object appearances.

3. Dataset Description

To evaluate FSL for UXO classification we required a broad collection of real sonar images that captures sensor type, seabed condition and object shape variability, a validation split that mirrors the five UXO categories of interest, and an out of distribution test set that represents the operational scenario. We therefore merged four publicly available SSS/forwardlooking sonar (FLS) datasets (SeabedObjects-KLSG [12], Marine Debris (MD) [13], Portuguese Navy (PN) [14], and Marine PULSE (MP) [15]).

Early experiments showed a strong correlation between object coverage and few-shot performance. Thus, the images were cropped from the source datasets using the provided bounding box annotations. The cropped image sizes within these datasets range from 43×42 pixels (smallest, drink carton, Fig. 1) to 2374×3092 pixels (largest, ship, Fig. 2).

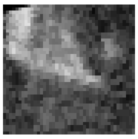


Fig. 1. Drink carton from Marine Debris dataset [13].

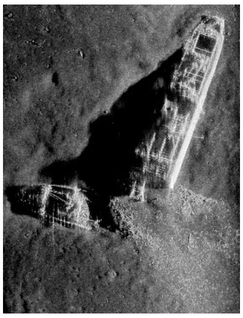


Fig. 2. Ship from SeabedObjects-KLSG [12].

To mirror the five UXO categories of interest, five classes were assigned to the validation set and the majority of the data to the training set. This approach maximizes the model's learning about representations of object characteristics, particularly mine-like contact (MILCO) and Non-mine-like bottom object (NOMBO) objects, which most closely resemble the target UXOs. The final corpus comprises 3734 training images and 1075 validation images (Table 1), spanning 13 object types and two sensor modalities.

The proprietary IRAV test set (Table II) was recorded in the Baltic Sea using an Edgetech 2205 SSS [16] and annotated using Sonarwiz. Objects were categorized by height, width, and length. Notably, all objects in the test set belong to the MILCO or NOMBO class found in the training set (represented by objects from the PN [14] dataset). The set comprises five object categories, each represented by 20 sonar images sourced from the internal IRAV collection: barrels, big elongated boxes, elongated boxes, small targets, and square boxes.

The SASOptical test set is a publicly available synthetic aperture sonar (SAS)/optical dataset (10 cylinders, 10 mantas, 14 naturals). It is used solely for baseline comparisons (PCA vs. YOLO-v8m-cls vs. FSL) because it is fully reproducible. It is not employed in the k-shot experiments due to its limited class count.

Table 1. Training and validation set.

Object type	Number of images	Original dataset	split
Bottle	449	MD [13]	train
Can	367	MD [13]	val
Chain	226	MD [13]	val
Drink carton	349	MD [13]	train
Engineering platform	82	MP [15]	val
Hook	133	MD [13]	train
MILCO	356	PN [14]	train
NOMBO	227	PN [14]	train
Airplane	56	KLSG [12]	train
Propeller	137	MD [13]	train
Seabed surface	88	MP [15]	val
Seafloor	578	KLSG [12]	train
Shampoo bottle	99	MD [13]	train
Ship	487	KLSG [12]	train
Standing bottle	55	MD [13]	train
Tire	321	MD [13]	train
Underwater residual mound	134	MP [15]	train
Valve	208	MD [13]	val

4. Methodology

Training FSL models with the dataset described in Section 3, we expect the models to learn to focus on highlight shadow contrast, object geometry, and texture continuity while minimizing the influence of sensor-specific noise.

4.1. Backbone Rationale and Modification for FSL

The YOLOv8m-cls backbone was utilized solely as a feature extractor, with the resulting 768-dimensional embedding (obtained after global average pooling and flattening) serving as input to the FSL models. It was chosen for its capacity to extract hierarchical features effectively capturing both local details and global contextual information. Initially designed for real-time object detection, the classification adaptation leverages a CSPNet-inspired architecture with cross-stage partial connections via C2f modules, optimizing computational efficiency while preserving gradient flow and feature diversity. This efficiency (low per-episode cost) is critical for FSL (meta-training is episode rich), enabling generalization from limited labelled data. ImageNet pretraining the YOLOv8m-cls backbone provides a robust initialization via transfer learning, adapting universal low-level features (e.g. edges, textures) to the target sonar domain.

4.2. Backbone Rationale and Modification for FSL

Our meta-learning training procedure comprises 5k-50k episodes for training (until each model converged). Metavalidation was performed using 500 additional episodes, each utilizing five support shots and 15 query images per class (a common benchmark setup [17, 18]). To ensure reproducibility, all experiments were conducted with a fixed random seed, which was used to randomly select 20 images per object from the meta-training/meta-validation set for each episode, and we used a batch size of one episode. All experiments were run on a workstation equipped with a 512 GB storage drive, an AMD EPYC 9554 processor, an NVIDIA L40S graphics card, and 46 GB of GPU memory.

a) Comparison FSL models: Based on Section 2, we evaluate four different FSL methods (DSN, ProtoNet, MatchingNet, and RelationNet) and compare five backbones (Conv32f [2], Conv64f [2], ResNet12 [18], ResNet18 [17], and YOLOv8m-based) to determine the best combination for real-world sonar images.

b) Baseline Comparison: After identifying the top FSL models for our UXO use case, we want to quantify how much these FSL configurations improve over two simple reference methods. First, PCA, a naive dimensionality reduction baseline that uses no learned representation. Second, a fine tuned YOLOv8m-cls with its original classifier head, presenting a standard supervised transfer-learning approach. We run all models (PCA, fine-tuned YOLOv8m-cls, DSN, ProtoNet) on the two tests presented in Section 3. The IRAV test set is chosen because it represents our target operational scenario. While the SASOptical dataset enables full reproducibility of the experiments. By evaluating the same models on both datasets we isolate the effect of the FSL head relative to the supervised baseline, and verify that the observed performance gains hold across a realistic UXO use case and a publicly accessible benchmark. By contrasting the two results, we directly measure how much improvement is achieved by introducing the FSL paradigm.

c) K-Shot Learning: To assess the data-efficiency of our few-shot approach and identify the smallest amount of labeled data that still yields reliable detection, we measured the performance of the best FSL model with support sets ranging from 1-shot to 8-shot, aiming to determine the minimum number of examples required to exceed a 60 % accuracy threshold on the challenging IRAV test set.

4.3. Evaluation Metrics

We evaluated the models using accuracy and F1-score. Disjoint class splits were enforced for meta-training, meta-validation, and meta-testing, ensuring no overlap and adhering to FSL principles. The validation classes are absent from both the training and test sets (class distributions in Tables 1). Robustness was measured by running 100 test episodes per set, reporting the average accuracy as the primary metric for limited-sample performance.

5. Results

This section presents the results of our experiments evaluating the performance of the proposed framework, exclusively on real-world sonar data.

a) Comparison FSL models: Tables 2 and 3 present the performance comparison across four different feature extractors, evaluated on the meta-validation set and the IRAV test set, respectively.

Table 2. Performance comparison of different feature extractors on validation set (F1-score in %).

Base	DSN	MatchingNet	ProtoNet	RelationNet
Conv32f	92.28	65.63	86.42	92.07
Conv64f	90.78	67.81	88.51	91.25
ResNet12	88.57	76.99	86.94	90.35
ResNet18	86.98	64.98	83.39	89.17
YOLOv8m	93.81	21.10	91.10	91.45

Table 3. Performance comparison of different feature extractors on IRAV test set (F1-score in %).

Base	DSN	MatchingNet	ProtoNet	RelationNet
Conv32f	61.80	15.35	54.88	54.02
Conv64f	63.62	25.99	57.47	57.87
ResNet12	67.89	20.36	60.00	14.34
ResNet18	53.09	21.67	54.22	11.43
YOLOv8m	77.21	19.89	83.89	33.95

The results make several important points clear. First, features extracted from the YOLOv8m classificator consistently outperform those from the traditional Conv and ResNet backbones, both on the meta-validation split and on the out of distribution IRAV test set, which underscores the benefit of using a backbone specifically designed for classification.

Second, the relative performance of FSL depends on the data split. The DSN classifier attains the highest scores on the validation set, whereas ProtoNet when paired with YOLOv8m features achieves the strongest results on the IRAV test set, indicating that ProtoNet is more robust to the domain shift between different sonar sensors. Third, MatchingNet proves unreliable across all backbones, delivering F1 scores between roughly 15 % and 28 %. This likely stems from its attention mechanism struggling with the noisy, low-contrast textures typical of sonar imagery. Fourth, RelationNet shows a pronounced overfitting tendency. It reaches competitive scores on validation but its performance collapses on the test split (F1 below 35 %), signaling poor adaptability to the domain gap. Taken together, these observations highlight that the combination of YOLOv8m-derived features with the prototype-based ProtoNet learner offers the most balanced and robust solution for practical sonar deployment, providing high accuracy on the validation stage while maintaining the best generalization when confronted with the challenging IRAV recordings.

b) Baseline Comparison: We first examined whether the FSL pipeline yields a tangible advantage over two conventional baselines (a fully fine-tuned YOLOv8m-cls classifier and a PCA analysis) on two sonar test sets (Table 4).

Table 4. Performance comparison on different SAS Datasets (Accuracy in %).

Tested Network	IRAV	SASOptical
PCA	60.00	69.20
YOLOv8m-cls	26.15	40.00
DSN with Conv32f	62.01	67.43
DSN with YOLOv8m	77.10	49.62
ProtoNet with ResNet12	61.23	61.29
ProtoNet with YOLOv8m	83.88	52.71

On the IRAV dataset ProtoNet equipped with the YOLOv8m backbone attained the highest accuracy (83.88 %). In contrast, on SASOptical the PCA approach achieved the highest overall accuracy (69.20 %), while all FSL configurations remained below 53 %.

A deeper inspection of class wise results explains where this advantage originates. On IRAV, ProtoNet+YOLOv8m attains >95 % accuracy on the most distinct “barrel” and “small-target” classes and maintains >70 % on the elongated and square boxes, while PCA never exceeds 66 % on any class. The large per class gains translate directly into the high overall accuracy reported in Table 4.

Conversely, on the SASOptical dataset PCA reaches 69.20 % overall because it perfectly classifies the two dominant “manta” and “natural” categories (85.7 % and 90.9 %). Its failure on the minority “cylinder” class (25.8 %) is compensated by the other two classes, whereas the few-shot models distribute their performance more evenly across all three categories, resulting in a lower average.

These findings demonstrate that the observed discrepancy between the two test sets is not an artefact of random variation but reflects genuine differences in class difficulty and domain characteristics. Consequently, for heterogeneous sonar datasets such as IRAV, prototype based FSL with a classification pretrained backbone is the preferred strategy, whereas for simpler, well separated datasets PCA may remain a viable low cost alternative (running the PCA took us a few minutes while each FSL model took several hours to train).

c) K-Shot Learning: To assess the data-efficiency of our few-shot approach and determine the smallest labeled support set that still yields reliable detection, we measured the performance of the best FSL model with support sets ranging from 1-shot to 8-shot, aiming to exceed a 60 % accuracy threshold on the challenging IRAV test set. We selected the ProtoNet+YOLOv8m configuration for this analysis because it achieved the highest performance in the initial Comparison FSL models experiment (Table 4).

Fig. 3 shows the per class classification accuracy (averaged over 100 independent episodes) as a function of the number of support shots. Accuracy rises monotonically for all classes, confirming that additional support examples improve both performance and stability. The standard deviation across classes drops from 6.74 % (1-shot) to 3.94 % (8-shot), indicating more consistent predictions as k increases, while the accuracy curve gradually flattens and converges toward its maximum.

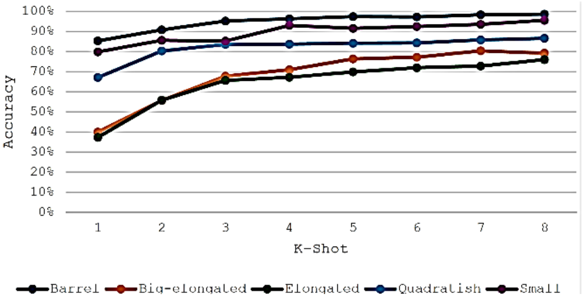
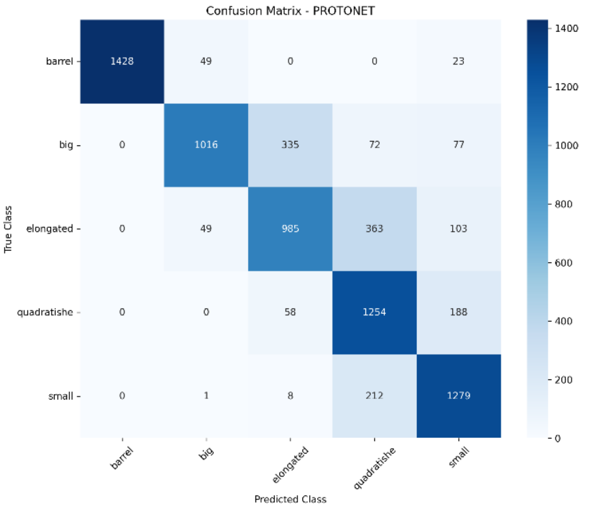


Fig. 3. Overview accuracy through increase of shots.

The barrel class achieves the highest performance: 85 % accuracy with a single shot and nearly 98 % with five shots, thanks to its distinct, large shape. In contrast, the big-elongated box and elongated box start below 40 % at 1-shot, only exceed 60 % after three shots, and reach just over 70 % at six shots, reflecting their visual similarity. The 3-shot confusion matrix (Fig. 4) highlights this issue: 22 % of big-elongated boxes are mislabeled as elongated boxes, and 24 % of elongated boxes are confused with square boxes.

Most of the accuracy gain occurs early in Fig. 3. The jump from 1 to 3 shots accounts for the majority of the improvement, after which the curve flattens, suggesting diminishing returns for additional support examples. Thus, the 3-shot setting already surpasses the 60 % threshold for all classes, demonstrating that only a few labeled examples are required for reliable detection in this domain.

For operational use in dumped-munition surveys, this rapid learning is advantageous because correctly identifying a few instances can quickly propagate to the remaining objects. Nevertheless, the residual confusion among the box-shaped categories suggests that integrating a confidence threshold or a secondary verification step would be beneficial for reducing false positives.



True Class \ Predicted Class	barrel	big	elongated	quadratishe	small
barrel	1428	49	0	0	23
big	0	1016	335	72	77
elongated	0	49	985	363	103
quadratishe	0	0	58	1254	188
small	0	1	8	212	1279

Fig. 4. Confusion matrix IRAV test set (100 episodes 5-way 3-shot).

6. Conclusion

The presented study demonstrates that FSL can substantially alleviate the data scarcity problem that hampers automated underwater object detection and classification, especially in legacy object rich environments such as UXO fields. By training exclusively on real world sonar imagery (including both SSS and FLS modalities) and by employing a YOLOv8m-cls backbone as a universal feature extractor, the proposed framework achieves a markedly higher level of discrimination than conventional baselines. On the meta-validation split DSN yields the best overall scores, yet on the out of distribution IRAV test set the prototype based ProtoNet combined with YOLOv8m-derived embeddings outperforms all other configurations (83.89 % F1), confirming that prototype learning is more robust to sensor and domain shifts. In contrast, MatchingNet consistently fails across all backbones, and RelationNet shows strong validation performance but collapses on the test data, indicating a tendency to overfit to the training distribution.

Baseline experiments confirm the benefit of the FSL pipeline: A fine-tuned YOLOv8m-cls reaches only 26 % accuracy on realistic IRAV recordings, a simple PCA-based classifier 60 %, both far below the 84 % achieved by ProtoNet+YOLOv8m. The PCA method stays competitive on the easier SAS-Optical dataset, showing that sonar's difficulty drives method choice.

The k-shot study shows performance climbing steeply up to three support examples per class and then flattening; 3-shot episodes already exceed 60 % accuracy for every object. This rapid learning means operational surveys can obtain reliable detection after labeling just a few representative instances, a major advantage for time consuming UXO clearance missions. Remaining confusions among morphologically similar box-shaped objects suggest that confidence-based filtering or a secondary verification stage could further cut false positives.

Future research should focus on two fronts. First, enriching the meta-training episodes with sonar-specific augmentations (e.g., speckle noise simulation, angle-dependent shadow modeling) is expected to narrow the domain gap between public datasets and proprietary recordings. Second, hybrid architectures that fuse the global discriminative power of unsupervised dimensionality reduction (e.g., PCA or auto encoders) with prototype based few shot classifiers may combine the strengths observed on both test sets.

Acknowledgements

The research presented in this paper was supported by the Bundesministerium für Wirtschaft und Energie (BMWE) within the project Industrielle Räumung von Altlasten in Verklappungsgebieten (IRAV). Furthermore, generative AI tools were utilized to assist with code development and text drafting. All content has been critically reviewed, edited, and remains the sole responsibility of the authors.

References

[1]. M. Ochal, J. Vazquez, Y. Petillot, S. Wang, A comparison of few-shot learning methods for underwater optical and sonar image classification, in *Proceedings of the Global Oceans 2020: Singapore-U.S. Gulf Coast*, 2020, pp. 1-10.

[2]. J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in Advances in Neural Information Processing Systems 30 (I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, et al., Eds.), *Curran Associates, Inc*., Red Hook, 2017, pp. 4077-4087.

[3]. O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in Advances in Neural Information Processing Systems 29 (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, et al., Eds.), *Curran Associates, Inc*., Red Hook, 2016, pp. 3650-3658.

[4]. F. Sung, Y. Yang, L. Zhang, T. Xiang, et al., Learning to compare: Relation network for few-shot learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1199-1208.

[5]. P. Ji, T. Zhang, H. Li, M. Salzmann, et al., Deep subspace clustering networks, in Advances in Neural Information Processing Systems 30 (I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, et al., Eds.), *Curran Associates, Inc.*, Red Hook, 2017, pp. 24-33.

[6]. C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in Proceedings of the 34th International Conference on Machine Learning (ICML), 2017, pp. 1126-1135.

[7]. C. Zhang, Q. Cui, S. Ren, Few-shot medical image classification with MAML based on Dice loss, in *Proceedings of the 2nd International Conference on Data Science and Computer Application (ICDSCA)*, 2022, pp. 348-351.

[8]. F. A. Spanhol, L. E. Oliveira, C. Petitjean, L. Heutte, A dataset for breast cancer histopathological image classification, *IEEE Transactions on Biomedical Engineering*, Vol. 63, Issue 7, 2016, pp. 1455-1462.

[9]. N. C. F. Codella, et al., Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the International Skin Imaging Collaboration (ISIC), *arXiv*, 2019, arXiv:1902.03368.

[10]. L. Lu, X. Cui, Z. Tan, Y. Wu, MedOptNet: Meta-learning framework for few-shot medical image classification, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 20, Issue 3, 2023, pp. 1785-1795.

[11]. J. Deng, W. Dong, R. Socher, L. Li, et al., ImageNet: A large-scale hierarchical image database, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248-255.

[12]. SeabedObjects-KLSG-II repository, https://github.com/HHUCzCz/-SeabedObjects-KLSG-II/commit/4a72e299adfae9d1fca1308e0a4275e8794f7f6b

[13]. D. Singh, M. Valdenegro-Toro, The marine debris dataset for forward looking sonar semantic segmentation, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3741-3749.

[14]. N. P. Santos, R. Moura, G. S. Torgal, V. Lobo, et al., Side scan sonar imaging data of underwater vehicles for mine detection, *Data in Brief*, Vol. 53, 2024, 110132.

[15]. X. Du, Y. Sun, Y. Song, L. Dong, et al., Revealing the potential of deep learning for detecting submarine pipelines in side scan sonar images: An investigation of pre training datasets, *Remote Sensing*, Vol. 15, Issue 19, 2023, 4873.

[16]. EdgeTech, 2205: AUV/UUV/ROV/ASV/USV sonars, https://www.edgetech.com/product/2200-and-2205-auv-uuv-rov-asvusv-sonars/

[17]. H. Scott, S. O'Toole, E. Ryan, K. McGuinness, et al., Impact of base dataset design on few-shot image classification, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 306-322.

[18]. G. S. Dhillon, P. Chaudhari, A. Ravichandran, S. Soatto, A baseline for few-shot image classification, in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

(009)

Local Binary Pattern Based Self Tuning Spectral Clustering for Images

S. Alamoudi [1], A. Bugshan [1], H. Wei [2], S. Khattak [3], X. Hong [2] and M. Almutairi [4]
[1] SABIC Manufacturing Services, UK
[2] University of Reading, UK
[3] Cardiff Metropolitan University, UK
[4] Zakat, Tax, and Customs Authority, Saudi Arabia
Tel.: + 966 506647157
E-mail: alamoudisa@sabic.com

Summary: In this paper, we propose a novel self-tuning spectral clustering technique for image classification. Uniform and full-histogram Local Binary Pattern (LBP) are used as features, while modified affinity matrices based on four distance measures are tested. The experimental results indicate that uniform LBP features generally achieve higher accuracy than full-histogram LBP features. On the distance-measure side, Bhattacharyya distance-based affinity matrices achieved higher accuracy than other distance measures, especially for large image datasets.

Keywords: Clustering, Spectral clustering, Self-tuning spectral clustering, Local binary pattern, Distance measures.

1. Introduction

Over the last few decades, we have seen the explosive growth of multimedia data, including text, images, and videos. Clustering techniques have gained significant importance because they facilitate the analysis of unlabeled data by revealing underlying structure [1]. Due to its simplicity, k-means [2] is one of the most popular clustering algorithms. One drawback of traditional clustering techniques is that they make simple assumptions about density estimation. For example, the k-means algorithm assumes that clusters are spherical and of similar size. However, in practice, many datasets are not spherical or of similar size [3].

Spectral clustering (SC) [4] has emerged as a popular alternative clustering technique. It is based on the eigenvectors of a matrix derived from the distances between points. SC differs from conventional clustering in its flexibility and does not assume a fixed cluster shape. Hence, in SC, it is possible for distant but connected points to belong to the same cluster, while points that are less distant but not connected belong to different clusters. Consequently, SC could be more efficient for data with varying shapes and sizes. Similarly, in terms of computational complexity, SC is faster than conventional clustering for sparse data sets of up to several thousand data points. However, for larger datasets, both the computational complexity and the accuracy of SC decrease.

An affinity function can serve an important role in the success of an SC algorithm [1]. However, when using a Gaussian radial basis function (RBF) as the affinity function, it is difficult to choose and manually set an appropriate scaling parameter σ. Ng et al. [4] proposed a method to select σ.

However, it is time-consuming, and the resulting value of σ may not perform well for data with clusters that have different local statistics [3]. Zelnik-Manor et al. [5] proposed a self-tuning spectral clustering (STSC) algorithm to address this problem. In particular, they addressed the issue of multiscaling data points. Nie et al. [6] introduced a constrained Laplacian rank algorithm for graph-based clustering, in which the affinity matrix is set to approximate a predefined similarity matrix. However, as the clustering outcomes are directly influenced by this similarity matrix, the results can be sensitive to its specification.

The local binary pattern (LBP) operator describes each pixel in a grey image based on the relative greyscale value of its neighbouring pixels [7]. If the greyscale value of the neighbouring pixel is lower, the value is set to zero; otherwise, it is set to one. The histogram of local binary patterns, as a feature vector, has been successfully used in many computer vision applications, including image classification, facial recognition, and texture recognition. In this paper, the STSC algorithm [5], previously used to solve image segmentation problems, is extended to image clustering problems. Moreover, new affinity functions with local scaling properties have been introduced based on LBP [7] features using various distance measures in probability space [8]. We carried out extensive comparative experiments, which have shown that uniform LBP features generally achieve higher accuracy than full-histogram LBP features, and that Bhattacharyya distance-based affinity matrices achieve higher accuracy than other distance measures, especially on large image datasets.

2. Image Processing Using Local Binary Pattern

In this section, we first provide a brief overview of the LBP image histogram, which serves as the input feature for the image clustering pipeline in our proposed method. The LBP operator encodes the structures around each pixel by labeling the pixels with decimal values (called LBPs or LBP codes). The operator begins by evaluating a 3 × 3 grid, where the intensity of the central pixel is compared with that of its eight neighbouring pixels within the image region. If the central pixel has a larger value compared to any of its neighbouring pixels, a 0 is placed in the corresponding position in the 3 × 3 grid. Otherwise, a 1 is placed. At the end of this process, the 3 × 3 grid is filled with binary values (0 or 1) except at the central position. The binary values are then concatenated starting from the top-left entry. As a result, an 8-bit binary number is generated and then transformed into its decimal equivalent. When the LBP code is mapped to a gray level value, this results in an LBP image from all pixels. A problem with the basic LBP operator is that it uses small neighbourhood of size 3 × 3 pixels which cannot capture important features of large scale structures [7]. It is known that among the 2^P possible binary patterns, some patterns are more frequent than the others. Consequently, a subset of these 2^P binary patterns is typically employed to characterize image textures. The patterns, known as uniform binary patterns [9], are denoted as $LBP_{P,R}^{U2}$. A binary pattern is considered uniform if it contains at most two bitwise transitions (i.e., changes from 0 to 1 or 1 to 0). A notable advantage of uniform LBP is its simplicity: for a neighbourhood of eight pixels, the standard LBP yields 256 distinct labels, whereas uniform LBP reduces this number to $n = 58$.

The histogram of an LBP-encoded image represents the frequency of occurrence (and implicitly, the probability distribution) of each pattern label across the image. For an 8-bit grayscale image, the standard LBP histogram comprises 256 bins, whereas the uniform LBP histogram contains 58 or 59 bins, depending on the implementation. As a result, uniform LBP enables a reduction in the dimensionality of the histogram feature vector. In our experiments, we compared the performance of both standard (non-uniform) and uniform LBP features.

3. Proposed Algorithm

3.1. Affinity Matrix Based on Distance Measure and LBP

This work extends the STSC algorithm [5] to image clustering based on their LBP features. Let $x = [x_1, x_2, \dots x_n]^T$ and $y = [y_1, y_2, \dots y_n]^T$. The choice of a distance measure is important since it directly impacts how similarity (or dissimilarity) between feature vectors (e.g. LBP feature vectors) (x, y) is interpreted. Different distance measures capture different aspects of similarity. In addition to Euclidean distance (which measures the straight line distance between two points) [10], other distance measures (which may be more suitable for capturing similarity between histogram-based features like LBP) investigated in this work are the Chi-squared distance [12], Cosine distance [13], Kullback Leibler Divergence (KLD) [14], and Bhattacharyya distance [15].

In STSC algorithm [5], by calculating a local scaling parameter σ_i for each data point $\boldsymbol{x_i}$, the distance between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ as 'seen' by $\boldsymbol{x_i}$ is $\frac{d(x_i,x_j)}{\sigma_i}$, while the converse is $\frac{d(x_i,x_j)}{\sigma_j}$. As a consequence, the square distance d^2 can be calculated as $\frac{d(x_i,x_j)d(x_j,x_i)}{\sigma_i\sigma_j} = \frac{d^2(x_i,x_j)}{\sigma_i\sigma_j}$. As a result, one can write affinity as follows:

$$\hat{A}_{ij} = \exp\left(\frac{-d^2(x_i,x_j)}{\sigma_i\sigma_j}\right) \tag{1}$$

In the case of points i and j, a specific scaling parameter allows self-tuning of the distance between the two points based on the local statistics of the vicinity. The local statistics around the point x_i can be studied to select the local scale σ_i. A simple choice for σ_i is:

$$\sigma_i = d(\boldsymbol{x}_i, \boldsymbol{x}_K), \tag{2}$$

where $\boldsymbol{x}_K$ is the K^{th} neighbour of point $\boldsymbol{x}_i$. The choice of K depends on the data dimension of the embedding space and it is independent of scale.

In this paper, it is proposed to perform clustering using local scaling based on a distance measure based on LBP features to improve the performance of the clusters. Unlike in the image segmentation problem; the local scaling relates to LBP images rather than points in an image. Moreover, it is proposed that automatic cluster generation be enabled, using only the parameter K, which is a function of the data dimension and is independent of scale. Traditionally, the distance d in Eq. (1) and Eq. (2) is computed using the Euclidean distance. However, probabilistic distance measures such as Chi-squared, Cosine, KLD, and Bhattacharyya is preferred for this purpose since LBP are histogram features. Let $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}] \in \Re^n, i = 1, \dots, N$ be a set of LBP feature vectors of N images, the relevant equations for these modified affinity matrix A at (i, j) are given below.

For Chi-squared distance measure, it is computed as follows:

$$A_{ij} = \exp\left(-\frac{\frac{1}{2}\sum_{l=1}^{n}\frac{\left(x_{i,l}-x_{j,l}\right)^2}{x_{i,l}+x_{j,l}}}{2\sigma_i^2\sigma_j^2}\right) \tag{3}$$

The cosine distance between two feature vectors x_i and x_j is defined as:

$$A_{ij} = \exp\left(-\frac{1-\frac{\sum_{l=1}^{n}x_{i,l}x_{j,l}}{\sqrt{\sum_{l=1}^{n}x_{i,l}^2}\sqrt{\sum_{l=1}^{n}x_{j,l}^2}}}{2\sigma_i^2\sigma_j^2}\right) \tag{4}$$

For the KLD distance measure, it is computed as follows:

$$A_{ij} = \exp\left(-\frac{\sum_{l=1}^{n} x_{i,l} \log\left(\frac{x_{i,l}}{x_{j,l}}\right)}{2\sigma_i^2\sigma_j^2}\right) \quad (5)$$

For the Bhattacharyya distance measure, it is computed as follows:

$$A_{ij} = \exp\left(-\frac{1-\sqrt{\sum_{l=1}^{n} \frac{\sqrt{x_{i,l}x_{j,l}}}{\sqrt{\left(\sum_{r=1}^{n} x_{i,r}\right)\left(\sum_{r=1}^{n} x_{j,r}\right)}}}}{2\sigma_i^2\sigma_j^2}\right) \quad (6)$$

3.2. The Proposed LBP-Based STSC Algorithm (Algorithm 1)

The proposed algorithm requires a set of input images to be clustered, from which their LBP histogram features are first extracted. The features act as a set of LBP operated points x_i which are provided at the input layer to the Affinity matrix and for distance measurement. In the next step, the local scaling parameter σ_i is computed for each point $x_i \in X$ using Eq. (2), where distance d is computed using different distance measures such as Chi-squared, Cosine Similarity, KL Divergence, and Bhattacharyya distance. This is followed by a step in which the locally scaled affinity matrix is formed. Then, a normalised affinity matrix L is constructed in the following step. Next, the C largest eigenvectors of matrix L are found.

Algorithm 1. The proposed LBP-based Self-tuning Spectral Clustering Algorithm for Image Classification

---

Require: input the data set of N images.

1. Obtain uniform and non-uniform LBP features using a rotation-invariant LBP operator. The feature represents a set of LBP-operated features as $X = \{x_1, x_2, \ldots, x_N\}$ in $x_i \in R^n$, that are required to be clustered.
2. Compute the local scale σ_i for each point $x_i \in X$ using (2) where distance d is computed using different distance measures such as Cosine Similarity, Chi-Squared, KL Divergence, and Bhattacharyya distance corresponding Eqns. (3)-(6) respectively.
3. Form the locally scaled affinity matrix $A \in R^{N\times N}$ where A_{ij} is defined according to (3)-(6) for $i \neq j$ and $\hat{A}_{ii} = 0$, respectively for various proposed distance measures.
4. Define D to be a diagonal matrix with $D_{ii} = \sum_{j=1}^{N} A_{ij}$ and construct the normalised affinity matrix $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$.
5. Find $y_1, \ldots, y_C$, the C largest eigenvectors of L and form the matrix $Y = [y_1, \ldots, y_C] \in R^{N\times C}$, where N is the number of data points while C is the largest possible number of clusters.
6. Use incremental gradient descent (see [5]) to find the rotation matrix R that best aligns Y's columns with the standard basis.
7. For each possible group number c, compute the alignment cost function according to (7). This evaluates how well the rotated eigenvectors align with cluster indicators.
8. Choose C_{best} as the largest c that gives the minimum alignment cost. This balances fitting the data well without overfitting with too many clusters.
9. Use the aligned matrix Z (top C_{best} eigenvectors after rotation) and assign point s_i to cluster c if and only if $max_j(Z_{ij}^2) = Z_{iC}^2$.
10. If the data is noisy, initialize k-means or EM clustering using the result of Step 9 as the starting point.

And then these eigenvectors are stacked in columns to form the matrix Y. This is followed by a step in which the rotation matrix R which best aligns Y 's columns with the canonical coordinate system is recovered. The costs of the alignment for each group number, up to C, are graded according to (7) (see [5]).

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} \frac{Z_{ij}^2}{M_i^2}, \quad (7)$$

where $Z = YR$ and $M_i = \max_j Z_{ij}$.

The final group number C_{best} is set to be the largest group number with minimal alignment cost. In the final step, the top C_{best} eigenvectors Z are taken and the original point x_i is assigned to cluster C if and only if $\max_j Z_{ij} = Z_{iC}^2$. In the case of highly noisy data, the result of the final step is used to initialise k-means or Expectation Maximisation (EM), clustering on the rows of Z.

4. Results and Discussion

Experiments were performed using four publicly available datasets: Flavia [16], KTH-TIPS2B [17], OuTEX TC 00013 [19], ALOT [20]. Details of these datasets are shown in Table 1.

Two experimental scenarios are being considered. In the first scenario, scaling/tuning is turned off (See Step 2, Algorithm 1). The idea here is to compare the results obtained using various distance measures. In the second scenario, to assess the impact of tuning, tuning is turned on. Fig. 1 – Fig. 7 show the results of our proposed STSC method. Results are shown when uniform and full histogram LBP features are used in unsupervised clustering. The horizontal axis shows the five different distance measures: Euclidean (1), Cosine

(2), Chi-squared (3), KLD (4), and Bhattacharyya distance (5) while the vertical axis shows the obtained accuracy.

Table 1. Details of the four datasets for validation experiments.

Dataset	Total Classes	Samples per Class	Total Samples	Sample Resolution
KTH-TIPS2B	11	16	176	100 × 100
ALOT	250	250	27,500	512 × 768
OuTEX TC 00013	68	20	1,360	128 × 128
Flavia	20	50	1,000	1280 × 960

In the self-tuning setting, no single global scaling parameter σ is manually selected. Instead, each data point x_i is assigned a local scale σ_i based on the distance to its K^{th} nearest neighbour, as defined in Eq. (2). In the reported experiments, K was set to 5, selected empirically after preliminary testing.

It can be seen from Figs. 1 – 4 that compared to full histogram LBP, uniform LBP features result in higher mean accuracy, especially when tuning is not used. This could be due to the fact that by reducing the dimensionality (from 256 bins to around 58 bins), uniform LBP avoids overfitting, especially when limited data is available. On the other hand, when tuning is used, full histogram LBP features also performed well. This could indicate that when tuning is optimized, richer features can be more useful. It is interesting to note that for both uniform and full histogram LBP features, Bhattacharyya distance achieves the highest mean accuracy. This could be due to the fact that LBP features are histogram based and Bhattacharyya distance measures similarity between probability distributions. On the other hand, other distance measures such as Euclidean and Cosine are less sensitive to underlying distribution of histogram features. Similarly, when tuning is used, it is clear that the accuracy of the algorithm increases significantly. Both tuning and the choice of a suitable distance measure like Bhattacharyya distance had a more significant impact for larger datasets like ALOT and Flavia.

Choosing the optimal value of K can be very challenging as it is not always straightforward, the best value depends on the data characteristics etc. The choice of K has a direct impact on the quality of classification results. In this paper, we chose K empirically.

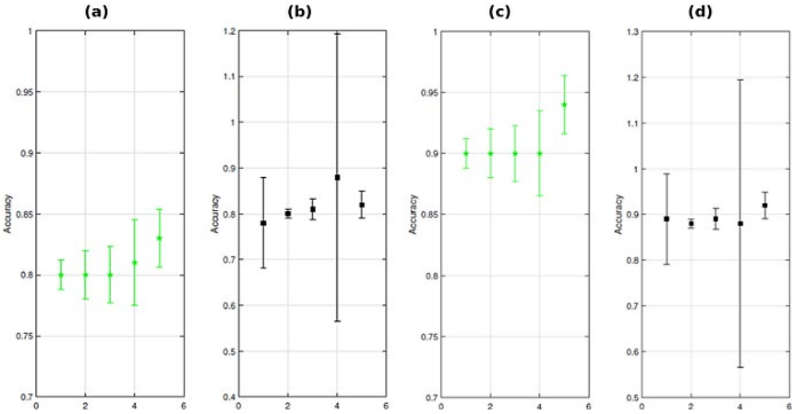


Fig. 1. Comparison of clustering accuracies for KTH TIPS2B dataset and five distance measures indexed by (1: Euclidean, 2: Chi-squared, 3: Cosine, 4. KLD, 5: Bhattacharyya). (a) uniform LBP features without tuning (b) full histogram LBP features without tuning (c) uniform LBP features with tuning, and (d) full histogram LBP features with tuning.

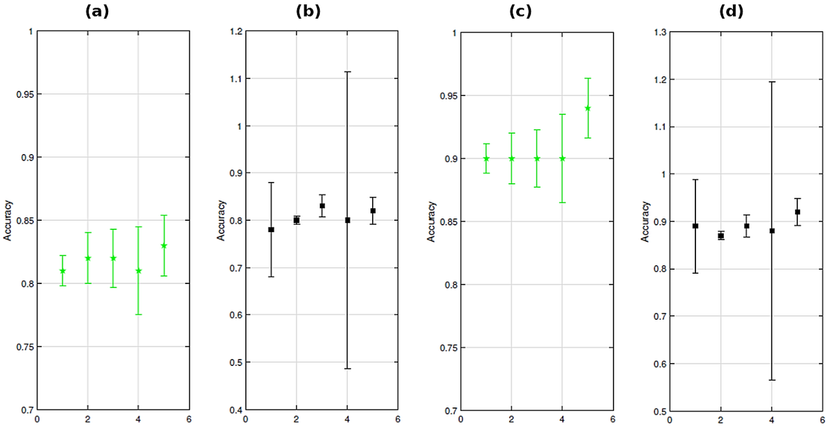


Fig. 2. Clustering accuracies for the Outex TC 00013 dataset and five distance measures indexed by (1: Euclidean, 2: Chi-squared, 3: Cosine, 4. KLD, 5: Bhattacharyya). (a) uniform LBP features without tuning (b) full histogram LBP features without tuning (c) uniform LBP features with tuning, and (d) full histogram LBP features with tuning.

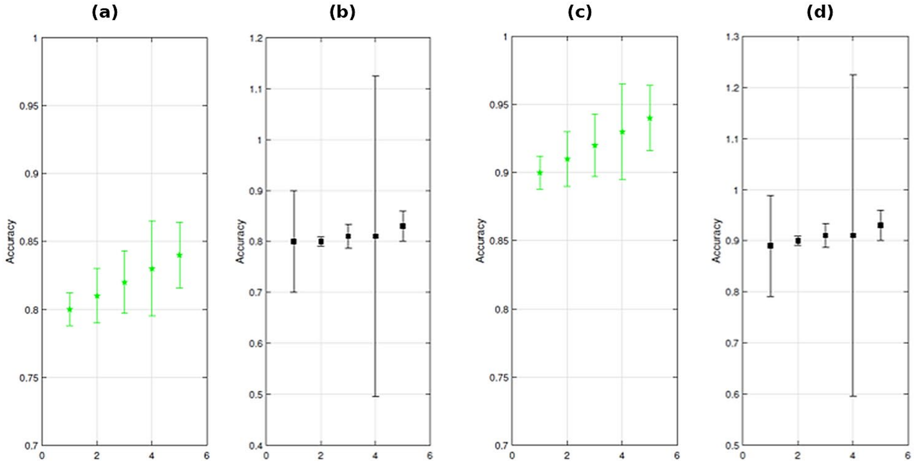


Fig. 3. Clustering accuracies for the Flavia dataset and five distance measures indexed by (1: Euclidean, 2: Chi-squared, 3: Cosine, 4. KLD, 5: Bhattacharyya). (a) uniform LBP features without tuning (b) full histogram LBP features without tuning (c) uniform LBP features with tuning, and (d) full histogram LBP features with tuning.

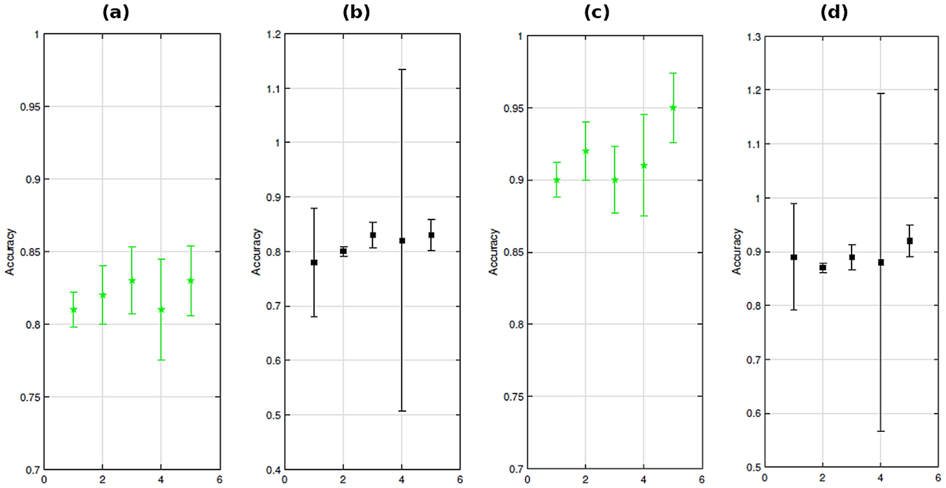


Fig. 4. Clustering accuracies for the ALOT dataset and five distance measures indexed by (1: Euclidean, 2: Chi-squared, 3: Cosine, 4. KLD, 5: Bhattacharyya). (a) uniform LBP features without tuning (b) full histogram LBP features without tuning (c) uniform LBP features with tuning, and (d) full histogram LBP features with tuning.

5. Conclusions

Self-tuning spectral clustering (STSC) addresses the problem of the multiscaling of data points in spectral clustering. It is based on the structure of the eigenvectors of an affinity matrix that represents pairwise data associations. Instead of selecting a single scaling parameter σ, a local scaling parameter σi is calculated for each data point. By analyzing the eigenvectors and finding the rotation of eigenvectors that creates the maximally sparse representation, the STSC technique can be used for automatic image segmentation. In this paper, STSC was extended to image clustering problems with LBP features and various distance measures. A set of input images was first used to generate LBP histograms. These histograms were then used as features. The affinity matrix technique was used for self-tuning spectral clustering. The proposed LBP-based STSC algorithms were evaluated for several different distance metrics. The experimental results indicate that uniform LBP features generally achieved higher accuracy than non-uniform LBP features. On the distance measure side, the Bhattacharyya distance measure outperformed the other distance measures.

References

[1]. G. Li, Z. Yu, K. Yang, M. Lin, et al., Exploring feature selection with limited labels: A comprehensive survey of semi-supervised and unsupervised approaches, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 36, Issue 11, 2024, pp. 6124-6144.

[2]. K. Sinaga, M. Yang, Unsupervised k-means clustering algorithm, *IEEE Access*, Vol. 8, 2020, pp. 80716-80727.

[3]. E. Schubert, Stop using the elbow criterion for k-means and how to choose the number of clusters instead, *ACM SIGKDD Explorations Newsletter*, Vol. 25, Issue 1, 2023, pp. 36-42.

[4]. A. Ng, M. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in Advances in Neural Information Processing Systems 14 (T. G. Dietterich, S. Becker, Z. Ghahramani, Eds.), *MIT Press*, Cambridge, 2002, pp. 849-856.

[5]. L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in Advances in Neural Information Processing Systems 17 (L. K. Saul, Y. Weiss, L. Bottou, Eds.), *MIT Press*, Cambridge, 2005, pp. 1601-1608.

[6]. F. Nie, Z. Zeng, I. Tsang, D. Xu, et al., Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering, *IEEE Transactions on Neural Networks*, Vol. 22, Issue 11, 2011, pp. 1796-1808.

[7]. J. H. Lin, J. Lazarow, A. Yang, D. Hong, et al., Local binary pattern networks, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 825-834.

[8]. S. Alamoudi, Image clustering and classification using content features, PhD Thesis, *University of Reading*, Reading, 2025.

[9]. Q. Zhao, Research on the application of local binary patterns based on color distance in image classification, *Multimedia Tools and Applications*, Vol. 80, Issue 18, 2021, pp. 27279-27298.

[10]. S. Zhang, J. Li, Y. Li, Reachable distance function for KNN classification, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, Issue 7, 2023, pp. 7382-7396.

[11]. Y. Rubner, C. Tomasi, L. Guibas, The earth mover's distance as a metric for image retrieval, *International Journal of Computer Vision*, Vol. 40, Issue 2, 2000, pp. 99-121.

[12]. W. Yang, L. Xu, X. Chen, F. Zheng, et al., Chi-squared distance metric learning for histogram data, *Mathematical Problems in Engineering*, Vol. 2015, Issue 1, 2015, 352849.

[13]. V. Prasatha, H. Alfeilate, A. Hassanate, O. Lasassmehe, et al., Effects of distance measure choice on KNN classifier performance – A review, *arXiv*, 2017, arXiv:1708.04321.

[14]. S. Kullback, R. Leibler, On information and sufficiency, *The Annals of Mathematical Statistics*, Vol. 22, Issue 1, 1951, pp. 79-86.

[15]. P. Reyes, J. Navarro, M. Santana, Comparative study of histogram distance measures for reidentification, *arXiv*, 2016, arXiv:1611.08134.

[16]. S. Wu, F. Bao, E. Xu, Y. Wang, et al., A leaf recognition algorithm for plant classification using probabilistic neural network, in *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2007, pp. 11-16.

[17]. B. Caputo, E. Hayman, P. Mallikarjuna, Class-specific material categorisation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005, pp. 1597-1604.

[18]. T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, et al., Outex – new framework for empirical evaluation of texture analysis algorithms, in *Proceedings of the 16th International Conference on Pattern Recognition (ICPR)*, 2002, pp. 701-706.

[19]. G. J. Burghouts, J. M. Geusebroek, Material-specific adaptation of color invariant features, *Pattern Recognition Letters*, Vol. 30, Issue 3, 2009, pp. 306-313.

(013)

A New Multimodal Co-Simulation Platform, Based on a Fleet of Small Autonomous Vehicles in a Containerized ROS2 Environment

M. Djoko-Kouam [1,2] **and A.-J. Fougères** [1]
[1] IT and Telecommunications Department, ECAM Louis de Broglie, Bruz, France
[2] Institute of Electronics and Digital Technologies of Rennes, CentraleSupélec, Rennes, France
Tel.: + 0033 299058454
E-mail: moise.djoko-kouam@ecam-ldb.fr

Summary: Autonomous vehicles today represent one of the major challenges for the future of the digital industry. From this perspective, the joint physical and digital simulation of these vehicles within their operational environment appears to us as a necessary step and support in the design of systems and processes in this field. In this article, we present a new multimodal platform for the co-simulation of a fleet of autonomous vehicles within their operational environment. The ROS2 environment in which the different autonomous entities of the system evolve is containerized in order to minimize deployment issues. We detail the design of a multi-level structure with a particular focus on floor-transition mechanisms, as well as the electronic, network, and communication design, thereby providing a clear overview of our co-simulator called COSiVA II. Use cases, particularly in the fields of airport and hospital logistics, provide a strong illustration of the relevance of the proposed approach and outline interesting perspectives for the continuation of our work.

Keywords: Autonomous vehicle, Co-simulation, Multimodal platform, Airport logistics, Hospital logistics, ROS2, Docker.

---

1. Introduction

Industry 4.0 aims to provide a framework, concepts and methodologies that enable a shift from mechanical or automated manufacturing production to digital production [1]. The key technologies of Industry 4.0 are [2]: the Industrial Internet of Things, Cloud Computing, Big Data, Simulation, Virtual and Augmented Reality, Additive Manufacturing, Horizontal and Vertical Systems Integration, Autonomous Robots and Intelligent robotic system, Cybersecurity.

One of the essential aspects of Industry 4.0 is the implementation of autonomous production methods carried out by robots, sometimes mobile, capable of intelligently performing tasks safely, flexibly, and potentially collaboratively. To achieve this, robots must become increasingly autonomous, flexible, and interactive in order to cooperate with each other and work safely alongside humans [3, 4].

In this context, simulation techniques can play a major role, enabling the evaluation of scenarios for the total or partial implementation of autonomous robotic systems. Indeed, simulation has two main objectives: 1) to develop planning and exploration models to optimize decision-making, design and operation of complex and intelligent (production) systems; 2) to facilitate the identification and assessment of risks, costs, implementation barriers, and the impact on operational performance of these complex and intelligent systems [5].

There are many simulation approaches applicable to Industry 4.0, mainly: Agent-Based Modelling and Simulation (*ABMS*), Discrete Event Simulation (*DES*), Virtual Reality (*VR*), Augmented Reality (*AR*), Artificial Intelligence (*AI*), Petri Nets (*PN*), Digital Twins (*DT*), and Hybrid Simulation (*HS*) [5].

ABMS is an approach increasingly used in industry for modeling and simulating complex systems composed of autonomous agents interacting with each other and with their environment [6, 7]. Thus, a strong analogy can therefore be drawn between a virtual agent and a physical (mobile) robot [8].

Co-simulation is an enhanced approach to simulation, where a physical simulation (often at a different scale) and a virtual simulation are coupled [9]. Beyond hybrid or digital twin simulations, the goal is to be able to analyze the behavior of a complex simulated system from both a physical and virtual perspective.

We have long used an initial simulation platform to simulate and analyze problematic scenarios involving the traffic of mobile mini-robots, and the modeling of this platform was the subject of a previous publication [10]. Following this work, we have recently designed a new co-simulation platform, named CoSiVA II. In the remainder of this article, we present its main architectural elements (physical, electronic, software, and communication), and propose three use cases: object localization, airport logistics simulation, and hospital logistics simulation.

2. Design of the Co-Simulation Platform

Over the years, different types of platforms have been designed and developed to test and experiment with autonomous navigation of mobile robots: in outdoor environments [11], in indoor environments [12], or on specific constructions with scaling that can accommodate micro-robots [13]. Mobile robots may be ground, aerial or underwater vehicles [14].

A co-simulation platform must necessarily consist of at least three architectural levels: one for physical simulation, another for virtual simulation, and a third to manage communication and interactions between the two previous levels. In the following sections, we present these three levels by considering four types of architectural components: physical, electronic, software, and communication.

2.1. Physical Architecture

Multi-level traffic area

The main architectural limitations of our previous platform were its lack of flexibility and genericity, both to adapt to new simulated environments and to increase its capabilities when the surfaces considered can vary in size, number and nature. The use case at the origin of the new co-simulation platform presented in this article was provided by our industrial partner as part of a collaborative project. This use case, which corresponds to an airport logistics scenario, consists of a first zone with 2 baggage pickup points and a second zone with 2 baggage drop-off points. The 2 zones are connected by a loop with a total length of 300 meters, corresponding to a realistic segment of a route taken by a piece of luggage during a transit stage or final delivery within an airport. Based on this industrial use case, the central idea of the platform is to transform a physical travel distance into a travel time within a multi-level circulation area. Fig. 1 provides an overview of the multi-level physical structure.

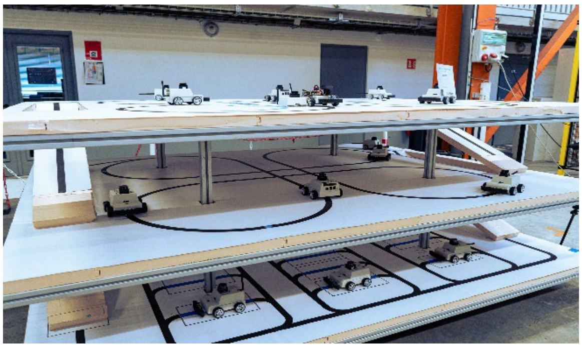


Fig. 1. Overview of the multi-level physical structure.

Level Transition Ramps

Our co-simulation platform, organized over three levels, has a total of four inter-floor transition ramps, which can be clearly identified in Fig. 1. Each ramp defines a three-section profile: an entry section, a smoothing section, and a linear slope section. The problem to be solved here is to determine the optimal curvature (rounding) required to smooth the slope, in order to allow our autonomous vehicle to enter and exit the ramp without either the underbody or the front bumper touching the ground. The optimality of this rounding determines the entry and exit crossing conditions of the ramp. The quality of the crossing depends on parameters related to the ramp profile, the vehicle underbody configuration, and the lower front skirt (also called the front bumper) of the vehicle. For the inter-floor transition ramp, it is necessary to take into account the vertical elevation change of the ramp, denoted H_s, as well as its horizontal span L_s. This indirectly incorporates the slope value, given by:

$$\alpha = \tan^{-1}\left(\frac{L_S}{H_S}\right) \tag{1}$$

Regarding the vehicle underbody, both its ground clearance H_v and the wheelbase of the vehicle (that is, the distance between the front and rear wheel axles) must be considered. For the lower front skirt of the vehicle, its height H_j must also be taken into account, together with the distance L_j between the front edge of the skirt and the axis of the front wheel. Fig. 2 provides a clear overview of the inter-floor transition ramp profile, along with the associated relevant parameters. All these parameters make it possible to dimension the central crossing component, referred to as the “slope smoother,” for its future fabrication using a 3D printer.

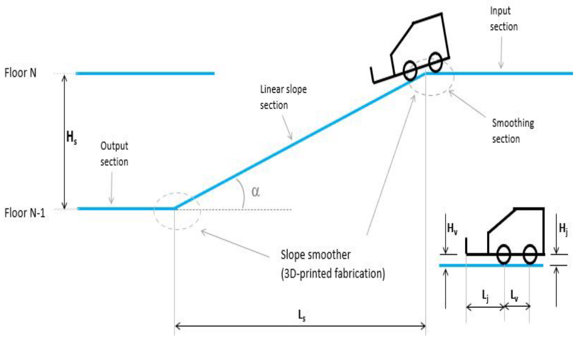


Fig. 2. Inter-floor transition ramp profile.

Thus, in order to prevent the vehicle underbody from touching the ground, the curvature of the transition between the horizontal section and the full slope must be sufficiently smooth so that the trajectory of the midpoint of the vehicle wheelbase always remains above H_v. This imposes a limitation on the maximum curvature of this transition. Overall, the slope smoother is defined by three parameters: its radius of curvature R, its arc length L_a, and its height H_a. The profile of the slope smoother is clearly illustrated in Fig. 3 for a slope variation of value α.

Let us consider the case corresponding to entering the ramp from the upper level. In Fig. 3, segment [E, B] represents the horizontal entry section at the top of the slope. Segment [C, D] corresponds to the full slope section. Finally, the arc (B, C), with radius R, represents the rounded profile of the slope smoother. Fig. 3 shows that the length of the smoother is given by:

$$L_a = R\alpha = R\tan^{-1}\left(\frac{L_S}{H_S}\right) \tag{2}$$

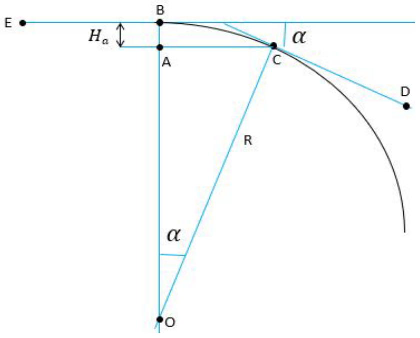


Fig. 3. Smoother profile for a slope variation of α.

$$H_a = R(1 - cos\alpha) \tag{3}$$

It can also be readily observed from the same figure that the height of the slope smoother is:

In Fig. 4 below, the vehicle chassis can be seen in a crossing position over the slope smoother, represented by the arc (H, L, G) of radius R. The anti-contact condition ensures that the midpoint M of the vehicle wheelbase always remains above point L on the road surface. In other words, the sagitta [K, L] of the smoother arc (H, L, G) must always remain less than or equal to the vehicle ground clearance H_v. A careful observation of Fig. 4 makes it possible to establish the expression of the sagitta F_c as a function of the radius R and the wheelbase L_v, according to the following formula:

$$F_c = KL = R - \sqrt{R^2 - \left(\frac{L_v}{2}\right)^2} \tag{4}$$

The anti-collision condition can therefore be expressed as:

$$F_c \leq H_v \tag{5}$$

A straightforward derivation of this condition makes it possible to express the minimum curvature radius of the slope smoother as:

$$R \geq R_{min}, with\ R_{min} = \frac{{L_v}^2}{8H_v} \tag{6}$$

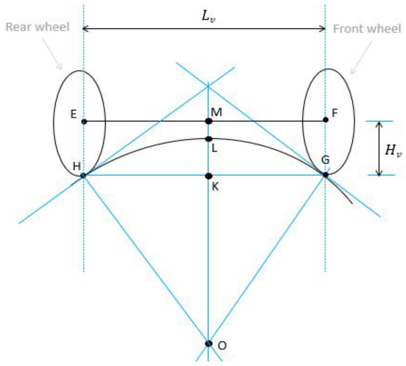


Fig. 4. Slope smoother and anti-contact condition.

The calculations presented here for our particular case are derived from theoretical developments that can be found in [15]. If we now consider the case corresponding to exiting at the bottom of the slope, it becomes easy to express the minimum curvature radius of the slope smoother as a function of the front bumper parameters, in the following form:

$$R \geq R_{min}, with\ R_{min} = \frac{{L_j}^2}{2H_j} \tag{7}$$

We observe that the minimum curvature radius of the slope smoother is not the same for the case of entering from the top of the ramp and that of exiting from the bottom of the slope. However, a 3D-printing project for the slope smoothers is more optimal when the parts share identical characteristics. From this perspective, and following a worst-case approach, we define a single minimum curvature radius for both situations according to the following formulation:

$$R \geq R_{min}, with\ R_{min} = max\left(\frac{{L_v}^2}{8H_v}, \frac{{L_j}^2}{2H_j}\right) \tag{8}$$

As a numerical illustration, the table shown in Fig. 5 below provides a summary of the numerical parameters of the slope smoother, calculated from the measured data of the inter-floor transition ramp, the vehicle, and the front bumper.

Ramp		**Vehicle**		**Front Bumper**		**Slope smoother**		
H_s	L_s	H_v	L_v	H_j	L_j	H_a	L_a	R_{min}
40	1000	13	95	13	85	20	107	280

Fig. 5. Calculated parameters of the slope smoother (all dimensions are in mm).

2.2. Electronic Architecture

Each autonomous vehicle in the fleet is an enhanced version of a commercial autonomous vehicle. Each vehicle is equipped with a steerable camera, a LiDAR, an *IMU* inertial module, and four independently motorized wheels. The choice of independently motorized wheels is motivated by the need for precise control of autonomous vehicle movements. The electronics of each autonomous vehicle are organized around a main board and two secondary boards (*ESP1* and *ESP2*). The main board is a Raspberry Pi 5 running at 2.7 GHz and equipped with 16 GB of *RAM*. Called *RPI5*, it acts as a host computer. It constitutes the functional core of the vehicle, ensuring the interface between internal components and external entities. We assign to it the control of a single sensor: the camera, due to its superior computing and memory capabilities. By default, the *RPI5* is associated with the *YB-EET01-V2.0* board equipped with an *ESP32-S3 WROOM-1* microcontroller. Called *ESP1*, this board includes four connectors for the wheel motors. The *IMU* module is also hosted on ESP1, and a dedicated connector allows a LiDAR to be connected. To extend testing capabilities, we added a second microcontroller board called *ESP2*, also based on an *ESP32-S3 WROOM-1*. This *ESP2* board is dedicated to additional sensors. It notably integrates an ultrasonic sensor for collision avoidance, an infrared *LED* strip for line tracking connected through a *PCF8574T* I2C module, as well as a *GPS* module enabling outdoor navigation and thereby materializing the multimodal nature of the co-simulation platform. Within the traffic area, certain positions must be detectable by each vehicle. *RFID* technology is used by equipping each vehicle with a reader and placing tags at each point of interest. When a vehicle passes over a tag within the correct speed range, the tag is detected. All sensors connected to the *ESP2* board use *GROVE* connectors through a “Grove shield,” providing a standardized 4-wire interface for any type of sensor. Fig. 6 presents the synoptic diagram of the electronic architecture of our autonomous vehicle model in its enhanced version adapted to our needs.

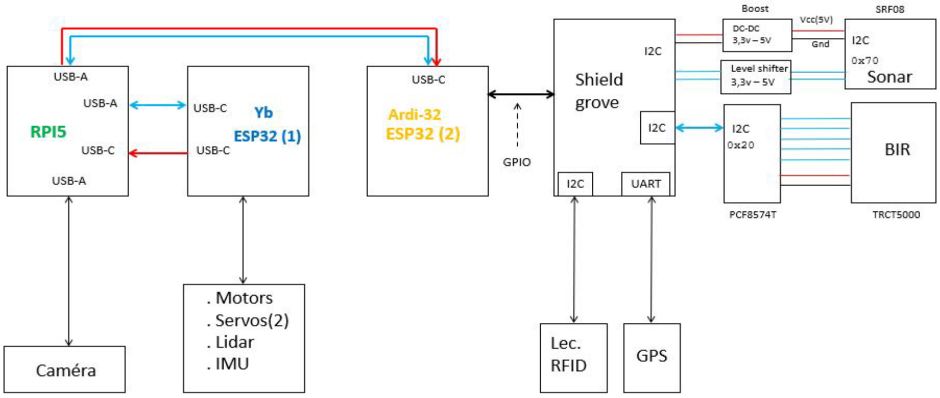


Fig. 6. Electronic architecture of the autonomous vehicle.

2.3. Software Architecture

Given the co-simulation nature of the platform, two types of software must be considered: on the one hand, software for the virtual component, and on the other hand, software enabling control of the physical component, including the operation of microrobots and their interactions. The virtual component is entirely developed in Python. It includes a supervisor, a user interface, and the instantiation of a potentially large number of fuzzy agents simulating the behavior of microrobots (either individually or as a fleet of robots). This architecture makes it possible to design scenarios in either centralized mode (continuous use of the supervisor) or decentralized mode (heavy reliance on communication between fuzzy agents to simulate microrobot autonomy) [16]. The software stack running on each of our autonomous vehicles is organized as follows: a first subset of autonomous components derived from Python models, executed on the *RPI5* board; a second subset of autonomous components (motors, *LiDAR*, *IMU*) derived from C/C++ models, executed on the first microcontroller board *ESP1*; and a third subset of autonomous components (notably sonar, infrared *LED* bar, *GPS*, *RFID* reader) also implemented in C/C++ and executed on the second microcontroller board *ESP2*.

The *RPI5*, as the main module, runs Ubuntu 20.04. In the home directory of the user pam-drone, we created a folder named *LD* containing a script called “install.sh”. Executing install.sh installs Docker, the container management system. The *LD* directory contains a subfolder RPI describing the creation of the container in which the autonomous components associated with the physical devices connected to the *RPI5* will run, notably the camera, but also autonomous components not associated with physical devices such as PositionControl or microRosAgent. The *LD* directory also contains the subfolder esp1, which describes - based on a Dockerfile – the creation of the container in which the autonomous components acting as proxies for the physical devices connected to the *ESP1* board will run. Finally, in a recent development, the *LD* directory includes the subfolder esp2, which also describes – based on a Dockerfile – the creation of the container in which the autonomous components representing the third group of physical devices will execute, namely those connected to the *ESP2* board. The physical components associated with *ESP2* include the infrared *LED* bar (*BIR*), the sonar, the *GPS*, and the *RFID* reader. Fig. 7 provides a synoptic overview of the software stack hosted on each of our autonomous vehicles.

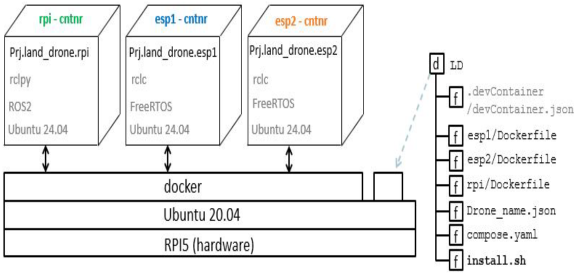


Fig. 7. Software stack hosted on each autonomous vehicle.

2.4. Communication Architecture

In the context of co-simulation, communication takes place at three levels: between fuzzy agents in the virtual world, between microrobots in the physical world, and between these two worlds, in either a centralized or decentralized manner. Let us recall that the electronic architecture of our autonomous vehicles integrates a Raspberry Pi 5 module acting as a host computer, as well as two *ESP32* modules serving as interface boards for sensors and actuators. Within this framework, we highlight a client/server communication model between the sensor interface boards (*ESP1* and *ESP2*) and the main board (*RPI5*). In practice, an autonomous component called a “micro-ROS agent,” running on the *RPI5*, is in continuous connection with another autonomous component called a “micro-ROS client,” running on each sensor interface board. This operational scheme makes it possible to create an effective unified *ROS2* network across all electronic boards, thereby providing each autonomous component with full visibility over published topics, regardless of the electronic board on which it is hosted.

From the perspective of the *ROS2* environment, the components of each autonomous vehicle are autonomous entities derived from the fundamental “Node” component defined in the *ROS2* framework. They communicate via Topics. The diagram in Fig. 8 illustrates the most relevant components of the autonomous vehicle from a communication standpoint.

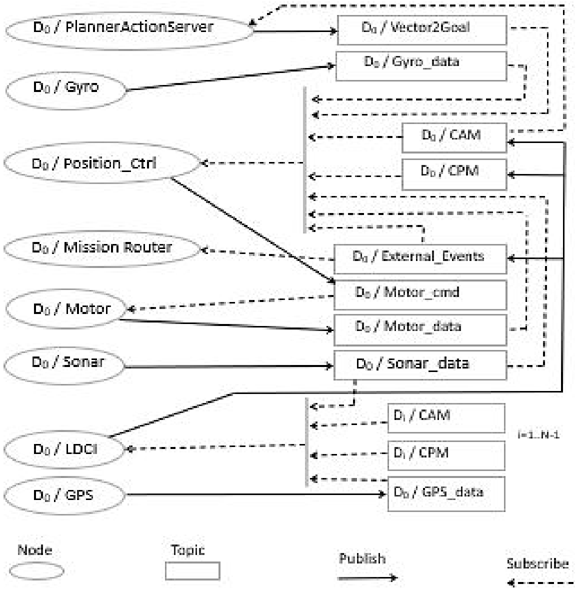


Fig. 8. Node and topic-based communication models of an autonomous vehicle.

For a proper understanding of this diagram, the following points should be emphasized: an oval represents a Node, a rectangle represents a Topic, a solid arrow represents a publication link, and a dashed arrow represents a subscription link. In this diagram, for example, the Sonar node publishes its data to the *Sonar_data* topic. It can also be seen that the motor command generated by the *D0/Position_Ctrl* node depends on various factors, such as obstacle proximity reported by the sonar. In order to fulfill its task of generating this motor command, the *D0/Position_Ctrl* node must subscribe to the appropriate topics, in particular *Sonar_data*. The Motor node, which is subscribed to the *Motor_cmd* topic, is notified whenever a new command is published on this topic. The execution of this command may result in the rotation of an electric wheel motor, which in turn generates odometry data. When started, the Motor node activates a timer, whose periodic trigger causes odometry data to be published to the *D0/Motor_data* topic. This data can then be used by any entity subscribed to this topic.

3. Some Application Use Cases

3.1. Localization of AGVs and Objects

The idea of AGV and object localization is oriented toward trajectory prediction and anticipation using a probabilistic approach based on Bayesian networks. This use case also allows us to extend the physical part of the co-simulation platform, by considering that our fleet of autonomous vehicles can behave like a fleet of ground drones capable of operating beyond their local deployment zone, thereby extending their operation to external environments, such as a car park area. This ability to deploy all or part of the autonomous vehicle fleet across two zones of completely different nature and scale illustrates the multimodal character of the co-simulation platform presented in this article.

3.2. Airport Logistics

The co-simulation platform proposed in this article is naturally well suited to airport logistics simulation scenarios. A first illustrative aspect of this adaptation is the multi-level structure, which reflects operational flows, including parking areas, charging zones, and loop-based circulation zones. A second illustrative aspect is the homogeneous nature of the fleet (25 identical vehicles), combined with unitary capacity and fine-grained task allocation defined by the correspondence 1 baggage item = 1 mission. Finally, a third illustrative aspect lies in the ability to test different fleet optimization strategies, including vehicle allocation, mission scheduling, and waiting time minimization, under combined objectives aimed, on the one hand, at reducing baggage delays and, on the other hand, at maximizing throughput.

3.3. Hospital Logistics

Our proposed co-simulation platform is also well suited to hospital logistics contexts. The first argument supporting this adaptability is the ease with which the multi-level structure can be transformed to represent a realistic hospital environment, in terms of medication storage areas, robot charging zones, and constrained circulation pathways. A second argument relates to the inclusion of human-related disturbances, such as the simulation of moving staff, stretcher-bound patients, mobile obstacles, or the random blockage of certain segments. Finally, a third argument concerns the management of medical priorities, based on the introduction of graded urgency levels (high, normal, low), for example linked to blood supply needs or critical medication delivery. These constraints may significantly impact scheduling, route replanning, and fleet management strategies.

4. Conclusion and Perspectives

We have presented in this article a new multimodal co-simulation platform based on a fleet of autonomous vehicles operating within a containerized *ROS2* environment. In its current version, the co-simulation platform was designed and its implementation began in the second half of the *ALPHA* project, which lasted 24 months. We plan to continue the development of this platform so that it can be used as a deployment testbed for airport logistics scenarios involving baggage pickup, transport, and delivery. In terms of future work, our laboratory is currently part of a consortium within a new collaborative project focusing on hospital logistics challenges. The overall objective of this new project is the development of Artificial Intelligence technologies that can be embedded in robots, drones, autonomous vehicles, and more broadly in intelligent objects (IoT) operating in a hospital environment. A key feature of this new project lies in the application of strong requirements at both conceptual and functional levels, particularly regarding the need for increased autonomy of these systems with respect to existing infrastructure, their ability to cooperate with third-party entities or their environment, and finally, the embedded cybersecurity of these systems. Our long-term goal is to achieve a co-simulation platform capable of operating across an even wider range of application domains, further increasing the multimodal nature of our co-simulation framework.

Acknowledgements

We thank the partners of the ALPHA project for their collaboration, which provided a supportive framework for this publication, as well as the Brittany Region and Rennes Métropole for funding our work (project certified by the Pôle Images & Réseaux, conv. 23003944).

References

[1]. E. Oztemel, S. Gursev, Literature review of Industry 4.0 and related technologies, *Journal of Intelligent Manufacturing*, Vol. 31, Issue 1, 2020, pp. 127-182.

[2]. V. Alcácer, V. Cruz-Machado, Scanning the Industry 4.0: A literature review on technologies for manufacturing systems, *Engineering Science and Technology, an International Journal*, Vol. 22, Issue 3, 2019, pp. 899-919.

[3]. M. A. K. Bahrin, M. F. Othman, N. H. N. Azli, M. F. Talib, Industry 4.0: A review on industrial automation and robotic, *Jurnal Teknologi (Sciences & Engineering)*, Vol. 78, Issue 6-13, 2016, pp. 137-143.

[4]. M. Soori, R. Dastres, B. Arezoo, F. K. G. Jough, Intelligent robotic systems in Industry 4.0: A review, *Journal of Advanced Manufacturing Science and Technology*, Vol. 4, Issue 3, 2024, 2024007.

[5]. W. de Paula Ferreira, F. Armellini, L. A. De Santa-Eulalia, Simulation in Industry 4.0: A state-of-the-art review, *Computers & Industrial Engineering*, Vol. 149, 2020, 106868.

[6]. C. M. Macal, Everything you need to know about agent-based modelling and simulation, *Journal of Simulation*, Vol. 10, Issue 2, 2016, pp. 144-156.

[7]. F. Michel, J. Ferber, A. Drogoul, Multi-agent systems and simulation: A survey from the agent community's perspective, in Multi-Agent Systems, *CRC Press*, 2018, pp. 17-66.

[8]. J. Grosset, A.-J. Fougères, M. Djoko-Kouam, C. Couturier, et al., Collision and obstacle avoidance for industrial autonomous vehicles – simulation and experimentation based on a cooperative approach, in Advances in Robotics and Automatic Control, Vol. 3 (S. Yurish, Ed.), *IFSA Publishing*, 2024, pp. 1-23.

[9]. M. Djoko-Kouam, A.-J. Fougères, Towards a bijective co-simulation model between physical and virtual environments, adapted to a platform for autonomous industrial vehicles, *Automation, Control and Intelligent Systems*, Vol. 11, Issue 2, 2023, pp. 27-44.

[10]. A. Ndao, M. Djoko-Kouam, A.-J. Fougères, Matrix beaconing for the location of autonomous industrial vehicles on a simulation platform, in *Proceedings of the 3rd International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI)*, 2021, pp. 80-86.

[11]. L. Wijayathunga, A. Rassau, D. Chai, Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review, *Applied Sciences*, Vol. 13, Issue 17, 2023, 9877.

[12]. C. Wang, L. Meng, S. She, I. M. Mitchell, et al., Autonomous mobile robot navigation in uneven and unstructured indoor environments, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 109-116.

[13]. N. Buckman, A. Hansen, S. Karaman, D. Rus, Evaluating autonomous urban perception and planning in a 1/10th scale minicity, *Sensors*, Vol. 22, Issue 18, 2022, 6793.

[14]. S. Nahavandi, R. Alizadehsani, D. Nahavandi, S. Mohamed, et al., A comprehensive review on autonomous navigation, *ACM Computing Surveys*, Vol. 57, Issue 9, 2025, pp. 1-67.

[15]. T. D. Gillespie, Fundamentals of Vehicle Dynamics, *SAE International*, Warrendale, 2021.

[16]. J. Grosset, O. Oukacha, A.-J. Fougères, M. Djoko-Kouam, et al., Fuzzy multi-agent simulation for collective energy management of autonomous industrial vehicle fleets, *Algorithms*, Vol. 17, Issue 11, 2024, 484.

(014)

Secure Model Context Protocol Architecture for Deep Data Loops

Jérémie Farret and Alexandre Ramtoula
Inmind Technologies, Inc. 3575 St Laurent Boulevard, Suite 200, Montreal, Quebec H2X 2T6, Canada
Tel.: +1-833-636-2269
E-mail: jfarret@inmindtechnologies.com

Summary: Large language models are increasingly used to access enterprise data, but many deployments still stop at read-only question answering. We present a secure Model Context Protocol (MCP) architecture for "deep data loops": governed workflows where datasets are queried, enriched and re-indexed as versioned data products. The design uses Retrieval-Augmented Generation (RAG) within a governed execution framework where tool access, policy, manifests and lineage are handled as part of the same control layer, so that multi-step tool execution remains safe and reproducible. A dedicated Enrichment MCP Server acts as a governed exposure layer for enrichment capabilities. This server exposes auditable operations while data stores are accessed via separate connector servers. We target sovereignty-sensitive deployments where privacy and auditability are central. Internal prototyping on approximately 10k time-series incident records illustrates a Retrieve → Enrich/Validate → Re-index loop, in which are retrieved daily narrative summaries derived from structured temporal records, representing a broader class of governed data products.

Keywords: Deep data loops, Model Context Protocol, Time-series RAG, Cyber resiliency.

1. Introduction

Retrieval-augmented generation (RAG) is a method to ground large language model (LLM) outputs in enterprise knowledge by combining parametric model memory with a non-parametric searchable corpus [1]. Yet many industrial deployments remain "read-only": retrieval is used to answer questions, but the system is not authorized (or not trusted) to execute governed data operations such as enrichment and write-back operations that would improve the data products over time. In environments with strict compliance and data sovereignty requirements, moving from "ask about data" to "operate on data" introduces hard requirements: strong access control for side effects, reproducibility, traceability, and resilience to security incidents. These concerns are reflected in recent Model Context Protocol (MCP)-specific security work, which frames the full tool-use lifecycle as a governance problem, from discovery and authentication to policy enforcement, provenance and auditability for tool-using LLM systems [2-4]. The contribution of this work is an architecture that applies MCP as a governed execution boundary for versioned Retrieve → Enrich/Validate → Re-index data loops.

We present an MCP-based architecture for governed deep data loops, allowing curated, versioned datasets as both LLM inputs and outputs of controlled tool execution. In this work, deep data refers to large-scale data transformed into actionable insight under quality safeguards. Time-series RAG serves as an entry point for temporal signal aggregation (e.g., anomaly detection from incident narratives), aligning with Information Technology (IT) signal processing for instance feeding into observability and cyber-resilient pipelines as well as Operational Technology (OT) signal processing for manufacturing 4.0 or smart supply chains.

2. Proposed Architecture

The proposed architecture make use of MCP as the boundary between orchestration and data operations. It keeps planning separate from the services that access and enrich data so that tool use remains auditable and easier to control. This separation is central to the design; the focus is on governing side effects when LLM-driven workflows move beyond read-only access. Following recent MCP threat analyses, the architecture also treats MCP server creation, deployment, operation, and update as security-sensitive lifecycle stages [4].

2.1. Architectural Components

Model Context Protocol is an open protocol intended to standardize how LLM applications connect to external data sources and tools [5, 6]. MCP defines a host-client-server pattern and a structured interface for capability discovery and tool invocation, providing a consistent integration layer across different systems. We use MCP as a way to connect systems, but also as a governed interface around which security checks, approvals, and audit logging can be applied before any data is modified. Compared with a conventional tool wrapper or ad hoc API integration, MCP provides a standardized boundary for tool discovery and invocation, and for policy enforcement across different systems. This makes it easier to reuse the same approval rules, access controls, and audit mechanisms across deployments instead of reimplementing them for each new integration. We do not position MCP as the implementation substrate for enrichment logic itself; rather, we use it as the governed interface through which selected enrichment capabilities are exposed to orchestration clients. This supports a reusable delivery pattern with shared governance contracts and project-specific extensions.

The proposed architecture is composed of (i) an LLM host with an agentic orchestrator responsible for planning and tool selection using an MCP client for tool and sources discovery, and controlled execution, (ii) one or more data connector MCP servers that expose read/write operations for managed stores, and (iii) a dedicated Enrichment MCP Server that provides an allow-listed catalog of transformation and enrichment tools. Because MCP clients rely on server-advertised tool names, descriptions, and schemas during discovery, these metadata fields are treated as security-sensitive artifacts: they are reviewed, versioned, and kept free of policy instructions that should instead be enforced server-side. MCP's specification describes the role separation between host, clients, and servers, and its standardized messaging model for tool connectivity [5]. The design intentionally separates orchestration (planning and sequencing) from capabilities (tools and sources), so governance stays consistent across different data sources and enrichment methods. This governance boundary is reflected in the proposed container structure.

Fig. 1 shows this separation at the container level. It summarizes how the Web User Interface (UI), LLM host with MCP client, policy and approval service, audit and lineage registry, and the MCP-based retrieval and enrichment services interact to support a governed Retrieve → Enrich/Validate → Re-index loop over the data store.

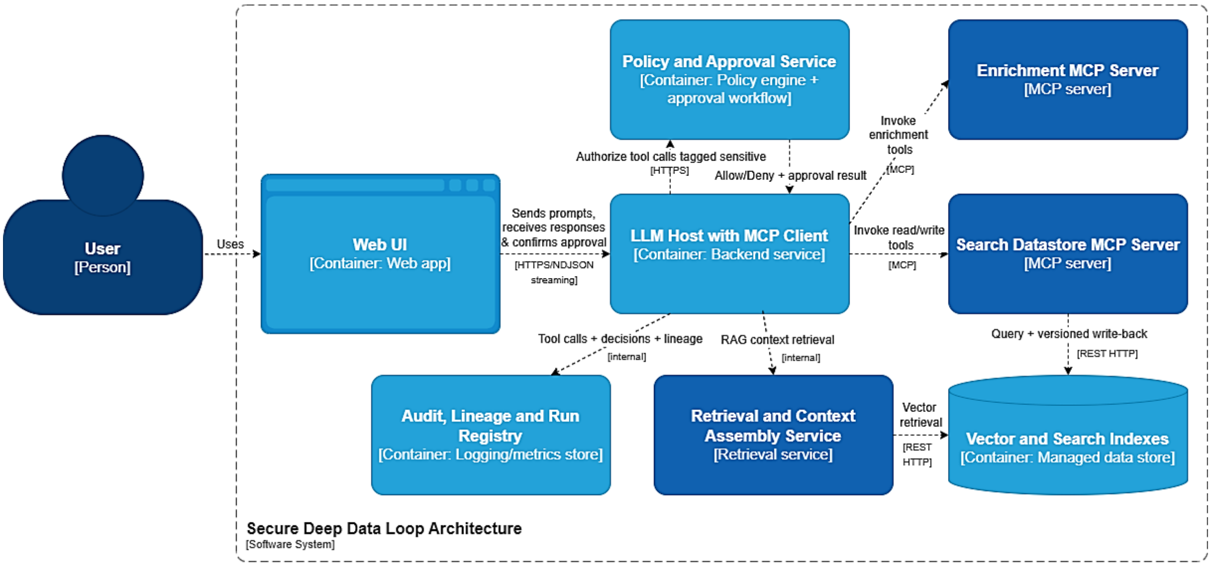


Fig. 1. Container diagram: MCP-orchestrated Retrieve-Enrich-Validate-Re-index loop.

2.2. Security and Resilience Controls

Security controls are applied at multiple layers. First, the MCP tool boundary defines a contract for each operation (inputs, outputs, and declared side effects), so policy checks and approvals can happen before any write occurs. Tool names and schemas are reviewed to avoid ambiguous or conflicting operations, and high-risk parameters such as destinations, file paths, index names, channels, and write targets are constrained through Enums, allow lists, regex validation, or server-side rejection of invalid arguments. Second, least privilege access is set separately for each tool and datastore role. This is reinforced by datastore-level security controls, such as controls that limit access to managed search services and indexed documents. These features help prevent unauthorized actions and help investigators understand what happened if an incident occurs [7]. Third, side-effect tools use safeguards to reduce the impact of errors and prevent unwanted large-scale changes. In deep data loops, these safeguards also include source-to-sink tracking so that data retrieved by one tool cannot be moved into another tool action without classification, minimization, redaction, destination checks, and audit logging. To align the design with cyber-resilience objectives, understood as the ability to resist and recover from cyber incidents, write-back uses a create-new-version-first approach: every enrichment run produces a new dataset version plus a run manifest capturing tool versions, parameters, timestamps, row counts, and traceability metadata, defined here as the links between source records, enrichment tools, generated artifacts, validation outputs, and resulting dataset versions. The same manifest supports objective validators, such as checks that no protected data reached an unauthorized sink, no enriched dataset was promoted without QA validation, and no high-impact write action occurred without approval. The LLM may support planning and candidate enrichment, but promotion depends on policy gates and validation tools and not the model judgment alone.

Internal prototyping on approximately 10k incident records was used to ground the architecture in a Retrieve → Enrich/Validate → Re-index scenario, from tool discovery and policy approval to enrichment, validation, versioned write-back, and manifest capture across successive dataset versions. Within the scope of this work, the prototype is used as an internal engineering reference point; evaluation of baselines, multiple-run statistics, latency overhead, and alternative orchestration approaches is left for subsequent work.

3. Governed Retrieval as the Read Path of Deep Data Loops

In the proposed architecture, RAG is included because it reflects how many enterprises already expose internal corpus data to LLM applications. In practice, RAG is often the standard pattern for question answering over internal knowledge bases. This extends RAG to time-series data: tabular incident records aggregate into daily narrative summaries (temporal signal aggregation for hybrid keyword-vector retrieval), enabling NetOps / SecOps signal processing like anomaly detection in cyber events, or enriching LLM generated summaries with temporal metadata, and can be indexed for hybrid retrieval.

In this architecture, retrieval is a controlled read path of the deep data loop. Daily summaries and temporal aggregates are indexed as versioned data products, so retrieval requests can be tied to dataset versions, access policies, and lineage metadata. Hybrid retrieval can still combine semantic, keyword, and temporal constraints [8], but its role is to provide traceable context before enrichment occurs. From an LLMOps perspective, retrieval configuration and evaluation artifacts become part of the same governance record as the downstream enrichment run.

For sovereignty-sensitive deployments, the architecture can run on controlled cloud, hybrid, or on-prem infrastructure, including Mind in a Box appliance when local execution is required. This deployment model supports tighter control over model serving, auditability, and confidential data exposure.

4. Conclusions

This architecture advances LLM-based data access beyond read-only data access, enabling auditable operations on versioned datasets via MCP-governed tools with RAG entry points. A prototype loop scenario on time-series incident records illustrates how versioned re-indexing supported by manifests and lineage metadata can support auditability and recovery-oriented workflows. Applicable to scheduled or streaming signal pipelines, it supports cyber-resilient intelligent systems in sovereignty-sensitive deployments.

Acknowledgements

The authors acknowledge Inmind Technologies for supporting this work and providing the internal engineering context used to ground the proposed architecture.

References

[1]. P. Lewis, E. Perez, A. Piktus, F. Petroni, et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, in Advances in Neural Information Processing Systems 33 (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, et al., Eds.), *Curran Associates, Inc.*, Red Hook, 2020, pp. 9459-9474.

[2]. H. Errico, J. Ngiam, S. Sojan, Securing the Model Context Protocol (MCP): Risks, controls, and governance, *arXiv*, 2025, arXiv:2511.20920.

[3]. X. Hou, S. Wang, Y. Zhang, Z. Xue, et al., SMCP: Secure model context protocol, *arXiv*, 2026, arXiv:2602.01129.

[4]. X. Hou, Y. Zhao, S. Wang, H. Wang, Model Context Protocol (MCP): Landscape, security threats, and future research directions, *arXiv*, 2025, arXiv:2503.23278.

[5]. Model Context Protocol specification, Version 2025-11-25, 2025, https://modelcontextprotocol.io/specification

[6]. Introducing the Model Context Protocol, Anthropic, 2024, https://www.anthropic.com/news/model-context-protocol

[7]. Data, privacy, and built-in protections in Azure AI Search, Microsoft Learn, https://learn.microsoft.com/en-us/azure/search/search-security-built-in

[8]. Vector search in Azure AI Search, Microsoft Learn, 2026, https://learn.microsoft.com/en-us/azure/search/vector-search-overview

(015)

Formal Paraconsistent Operators for Acoustic Feature Selection in Voice-Based Parkinson's Detection

M. E. Orlandini and R. C. Guido
São Paulo State University (Unesp), Ibilce, Cristóvão Colombo, 2265, São José do Rio Preto,
15054-000, Brazil
Tel.: +55 19 99307 6665
E-mail: mayaraeid@gmail.com

Summary: This work proposes a formal feature selection framework for Parkinson's disease (PD) detection from voice signals, grounded in the Two-Value Annotated Paraconsistent Logic (LPA2v). The method employs the formal operators min(μ) and max(λ) to combine multiple acoustic evaluation criteria under a conservative conjunction principle: a feature is selected only when all criteria simultaneously present sufficient favorable evidence and low contrary evidence. The certainty degree Gc ranks features without requiring empirically defined weights or thresholds. The framework is validated on two datasets with contrasting acoustic conditions – a laboratory-controlled Italian corpus and a telephone-quality recording set – achieving G-Mean of 0.901 and 0.728, respectively. Gc naturally encodes signal quality, distinguishing high-confidence laboratory recordings from degraded telephone samples without external metadata.

Keywords: Parkinson's disease detection, Paraconsistent logic, Feature selection, Acoustic biomarkers, Voice analysis, Interpretable machine learning.

---

1. Introduction

Acoustic analysis has established itself as a non-invasive and low-cost digital biomarker for Parkinson's disease (PD) detection and monitoring [1, 2]. Prior work has explored jitter, shimmer, and harmonic-noise-ratio features combined with statistical classifiers [3], including telephone-quality recordings [4]; systematic reviews confirm the breadth of acoustic approaches across protocols [5]. However, as the number of extracted features grows, feature selection becomes critical to reduce noise and prevent overfitting [6]. Current methods tend to be heuristics [6, 7], lacking formal logical consistency.

Paraconsistent logic has also been applied to voice and pathology analysis: the Discriminative Paraconsistent Machine (DPM) [8] introduced a trained decision surface over paraconsistent evidence scores for pattern classification; its acoustic extension [9] further applied DPM-derived scores to voice pathology screening. A Paraconsistent Feature Engineering (PFE) framework [10] later formalized evidence-based feature scoring for general signal classification, and an early PD voice screening system [11] combined acoustic features with machine-learning-based decision rules. Unlike these prior works, which either train a classifier over paraconsistent scores or apply heuristic ranking thresholds, the present approach uses the formal Two-Value Annotated Paraconsistent Logic (LPA2v) operators directly as the selection criterion, so that a feature is admitted only under a unanimous multi-criteria decision.

The LPA2v provides exactly such a framework, quantifying degrees of favorable (μ) and contrary (λ) evidence [12-14]. Inspired by this structure, this work proposes the use of the formal operators min(μ) and max(λ), corresponding to conservative conjunction over the paraconsistent lattice [14], to combine multiple acoustic feature evaluation criteria.

The proposed method is validated on the Italian [15-18] and Telephone [19-21] datasets, demonstrating that formal paraconsistent conjunction yields compact and interpretable feature subsets without dataset-specific tuning.

2. Methodology

Two datasets with contrasting acoustic conditions were used: the Italian Parkinson's Voice and Speech database [15] (28 PD, 22 elderly healthy control (HC), professional microphones; doi:10.21227/aw6b-tg17) and a telephone-quality recording set [19-21] (40 PD, 41 HC; doi:10.6084/m9.figshare.23849127), transmitted via telephone network.

From the Italian corpus, the 22 elderly HC were retained as an age-matched subset relative to the PD group, excluding 15 younger controls to avoid age-related acoustic confounds. For each subject, the sustained vowel /a/ (first repetition) was selected, following standard protocol in PD voice research.

Twenty-four acoustic features were extracted per recording across five groups: pitch, energy, spectral, harmonics-to-noise, and composite indices. For each feature, six evaluation criteria are computed: three evidences of support measures (μ) – a statistical separability test (KS), a stability criterion (Fisher), and single-feature G-Mean – and three evidences of contradiction measures (λ) – overlap coefficient (OVL), outlier ratio, and directional inconsistency.

In LPA2v [12-14], each feature is annotated by a pair $(\mu, \lambda) \in [0,1]^2$, where μ represents the degree of favorable evidence for class separability and λ the degree of contrary evidence. These values locate the feature on a paraconsistent lattice whose vertices represent True (V), False (F), Inconsistent (⊤), and Indeterminate (⊥). A feature positioned near V (high μ, low λ) is a reliable discriminator; near ⊥ (low μ, low λ) it is uninformative.

Formal paraconsistent operators (Eq. (1)-(2)) combine these criteria; features are ranked by certainty degree Gc (Eq. (3)) and the top-k = 3 selected, following an inner cross-validation sweep in which larger values degraded performance.

$$\mu = \min(\mu_{KS}, \mu_{fisher}, \mu_{g-mean}), \tag{1}$$

$$\lambda = \max(\lambda_{OVL}, \lambda_{outlier}, \lambda_{directional}), \tag{2}$$

$$G_C = \mu - \lambda \tag{3}$$

Selected features are normalized, direction-corrected, and combined as a μ-weighted average. Classification uses a threshold optimized to maximize G-Mean.

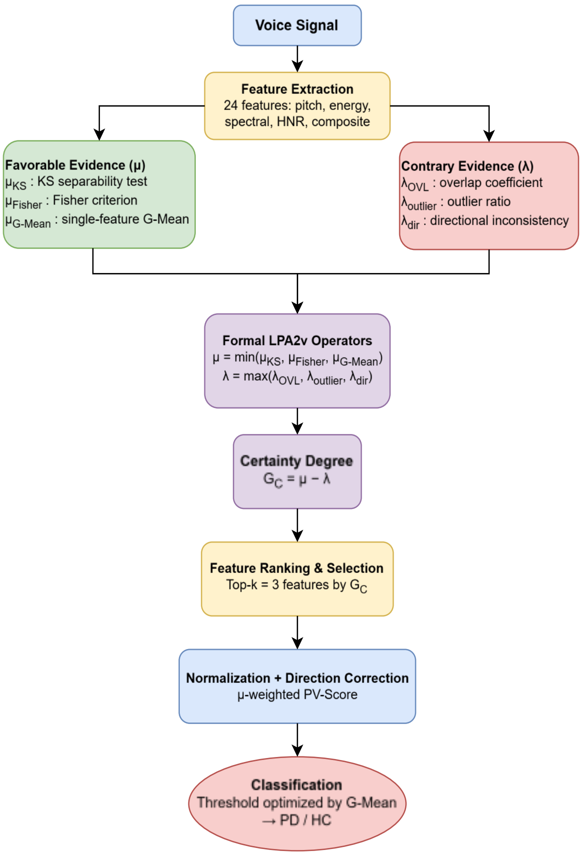


Fig. 1. Block diagram of the proposed LPA2v-based feature selection and classification pipeline.

3. Results

Table 1 summarizes the classification results under Leave-One-Out (LOO) and 5-fold cross-validation for both datasets.

Table 1. Classification performance.

Dataset	Evaluation	G-Mean	AUC
Italian	LOO	0.901	0.958
	5-fold	0.884 ± 0.165	0.964
Telephone	LOO	0.728	0.787
	5-fold	0.694 ± 0.124	0.741

The formal operators selected three features across all folds. For the Italian dataset (Fig. 2a) the selected features were mean energy (E_mean), energy coefficient of variation (E_CV), and energy asymmetry (E_asymm), and the Telephone dataset (Fig. 2b) energy standard deviation (E_std), energy coefficient of variation (E_CV), and spectral centroid coefficient of variation (SC_CV).

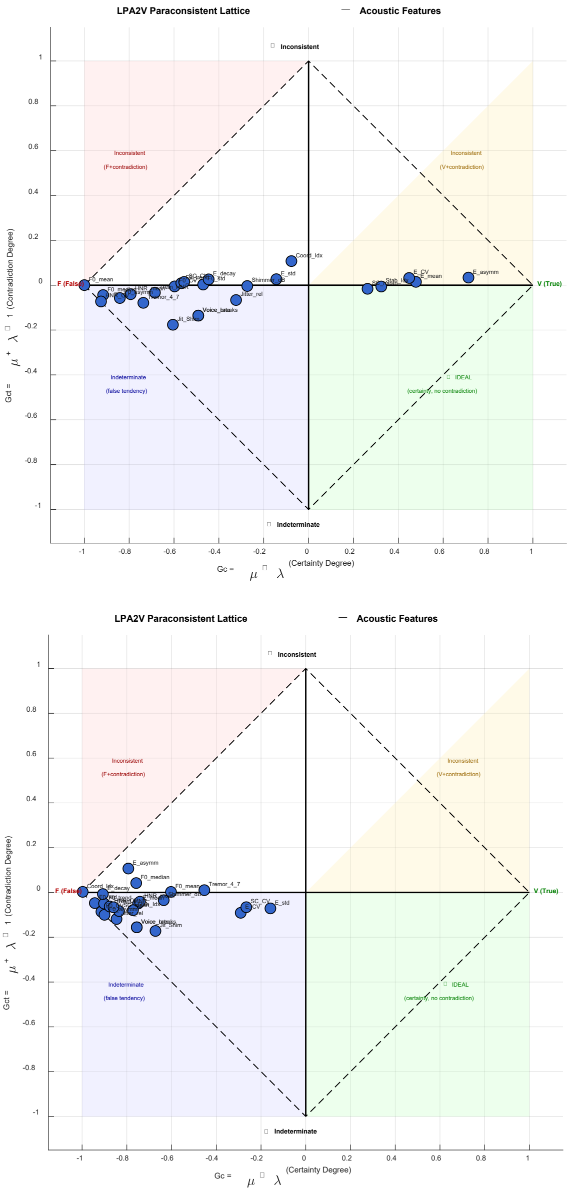


Fig. 2. LPA2v paraconsistent lattice for the (a) Italian and (b) Telephone datasets. Each point represents one acoustic feature.

All Italian samples were classified with high confidence (mean Gc = +0.55), whereas Telephone samples were classified with moderate confidence (mean Gc = −0.23). This contrast arises from lower μ (0.79 vs. 0.34) and higher λ (0.24 vs. 0.58) in telephone recordings, reflecting acoustic degradation captured without external channel metadata.

4. Discussion

The performance gap between datasets reflects differences in acoustic quality: telephone recordings yield consistently lower Gc, suggesting that LPA2v operators naturally encode signal degradation without external metadata. Feature selection stability across folds is explained by the non-compensatory operators: a feature is selected only when all criteria agree simultaneously. The shift from E_asymm/E_mean to E_std/SC_CV across datasets reflects acoustic adaptation rather than instability. Evaluation on larger corpora is planned.

5. Conclusion

This work proposes a formal paraconsistent feature selection framework for PD voice detection, in which min(μ)/max(λ) operators ensure features are selected only under unanimous evidence agreement. Results confirm compact, interpretable, and acoustically adaptive feature subsets. The LPA2v confidence score emerges as an intrinsic signal quality indicator, opening directions for protocol-aware PD screening.

References

[1]. F. Amato, G. Saggio, V. Cesarini, G. Olmo, et al., Machine learning- and statistical-based voice analysis of Parkinson's disease patients: A survey, *Expert Systems with Applications*, Vol. 219, 2023, 119651.

[2]. M. Rashidi, S. Arima, A. C. Stetco, et al., Prediction of Parkinson disease using long-term, short-term acoustic features based on machine learning, *Brain Sciences*, Vol. 15, Issue 7, 2025, 739.

[3]. L. Naranjo, C. J. Pérez, J. Martín, Y. Campos-Roca, A two-stage variable selection and classification approach for Parkinson's disease detection by using voice recording replications, *Computer Methods and Programs in Biomedicine*, Vol. 142, 2017, pp. 147-156.

[4]. S. Arora, A. Tsanas, Assessing Parkinson's disease at scale using telephone-recorded speech: Insights from the Parkinson's Voice Initiative, *Diagnostics*, Vol. 11, Issue 10, 2021, 1892.

[5]. Q. C. Ngo, M. A. Motin, N. D. Pah, P. Drotár, et al., Computerized analysis of speech and voice for Parkinson's disease: A systematic review, *Computer Methods and Programs in Biomedicine*, Vol. 226, 2022, 107133.

[6]. J. Li, K. Cheng, S. Wang, F. Morstatter, et al., Feature selection: A data perspective, *ACM Computing Surveys*, Vol. 50, Issue 6, 2017, pp. 1-45.

[7]. D. Theng, K. K. Bhoyar, Feature selection techniques for machine learning: A survey of more than two decades of research, *Knowledge and Information Systems*, Vol. 66, Issue 3, 2024, pp. 1575-1637.

[8]. R. C. Guido, S. Barbon Jr., R. D. Solgon, K. C. S. Paulo, et al., Introducing the discriminative paraconsistent machine (DPM), *Information Sciences*, Vol. 221, 2013, pp. 389-402.

[9]. E. S. Fonseca, R. C. Guido, S. Barbon Jr., H. Dezani, et al., Acoustic investigation of speech pathologies based on the discriminative paraconsistent machine (DPM), Biomedical *Signal Processing and Control*, Vol. 55, 2020, 101615.

[10]. R. C. Guido, Paraconsistent feature engineering [lecture notes], *IEEE Signal Processing Magazine*, Vol. 36, Issue 1, 2019, pp. 154-158.

[11]. S. Barbon Junior, V. G. T. Costa, S.-H. Chen, R. C. Guido, U-healthcare system for pre-diagnosis of Parkinson's disease from voice signal, in *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, 2018, pp. 271-274.

[12]. J. M. Abe, Fundamentos da Lógica Anotada (Foundations of Annotated Logics), PhD Thesis, *Universidade de São Paulo*, São Paulo, 1992 (in Portuguese).

[13]. J. I. Da Silva Filho, Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com Dois Valores LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos, PhD Thesis, *Universidade de São Paulo*, São Paulo, 1999 (in Portuguese).

[14]. J. I. Da Silva Filho, J. M. Abe, et al., Paraconsistent annotated logic algorithms applied in management and control of communication network routes, *Sensors*, Vol. 21, Issue 12, 2021, 4219.

[15]. G. Dimauro, F. Girardi, Italian Parkinson's voice and speech, *IEEE DataPort*, 2019, 10.21227/aw6b-tg17.

[16]. G. Dimauro, D. Caivano, V. Bevilacqua, F. Girardi, et al., VoxTester, software for digital evaluation of speech changes in Parkinson disease, in *Proceedings of the IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2016, pp. 1-6.

[17]. G. Dimauro, V. Di Nicola, V. Bevilacqua, D. Caivano, et al., Assessment of speech intelligibility in Parkinson's disease using a speech-to-text system, *IEEE Access*, Vol. 5, 2017, pp. 22199-22208.

[18]. L. Simone, M. G. Camporeale, V. M. Rubino, V. Gervasi, et al., Interpretable early detection of Parkinson's disease through speech analysis, in *Proceedings of the International Conference on Artificial Intelligence in Medicine (AIME)*, 2025, pp. 373-378.

[19]. T. Virmani, M. Lotia, A. Glover, L. Pillai, et al., Feasibility of telemedicine research visits in people with Parkinson's disease residing in medically underserved areas, *Journal of Clinical and Translational Science*, Vol. 6, Issue 1, 2022, e133.

[20]. F. Prior, Voice samples for patients with Parkinson's disease and healthy controls, *Figshare*, 2023, 10.6084/m9.figshare.23849127.

[21]. A. Iyer, A. Kemp, Y. Rahmatallah, L. Pillai, et al., A machine learning method to process voice samples for identification of Parkinson's disease, *Scientific Reports*, Vol. 13, 2023, 20615.

(016)

High Precision, Efficient Deep Learning Approaches for Embedded Fruit Quality Classification

M. A. Bouallegui [1,2], **A. Abadi** [2], **I. Saidi** [1] **and D. Fofi** [2]
[1] University of Tunis El Manar, Laboratoire de Recherche en Automatique, Tunis, Tunisia
[2] Université Bourgogne Europe, Laboratory ImViA EA 7535, University of Bourgogne,
21000 Dijon, France
Tel.: + 216 58 659 373
E-mail: aminebouallegui20@gmail.com

Summary: Automated fruit quality inspection plays a critical role in smart agriculture and Industry 4.0 by enabling efficient sorting, grading, and quality control while reducing reliance on manual inspection. In this paper, a comparative study of lightweight deep learning approaches for fruit quality classification is presented, with particular emphasis on real-time performance and embedded deployment. A custom convolutional neural network (CNN) with reduced parameter complexity is proposed to achieve fast training, low computational cost, and efficient inference while maintaining high classification accuracy. In addition, a transfer learning approach based on MobileNetV3 and a detection-oriented architecture, YOLO26n, are evaluated to investigate the trade-offs between different architectural paradigms. The dataset consists of multiple fruit categories, including fresh and defective samples, and is enhanced through preprocessing and data augmentation techniques to improve robustness and generalization. The proposed CNN integrates global average pooling and optimized hyperparameters to reduce overfitting and improve feature learning efficiency. Experimental results demonstrate that lightweight models can achieve high accuracy in distinguishing subtle variations in fruit quality while maintaining low computational requirements. Furthermore, the comparison highlights that detection-based architectures can be effectively adapted for classification tasks, providing competitive performance in resource-constrained environments. These findings confirm the potential of lightweight deep learning models for real-time agricultural applications on embedded systems.

Keywords: Fruit classification, Deep learning, Embedded systems, Lightweight CNN, TensorFlow, MobileNetV3, YOLO, Computer vision.

1. Introduction

Automated fruit quality inspection has become an essential component of modern smart agriculture and Industry 4.0 systems, enabling efficient sorting, grading, and post-harvest management. Ensuring high-quality agricultural products is critical for meeting market standards and reducing economic losses caused by defects, diseases, or improper ripeness. Traditional inspection methods, which rely heavily on human operators, are often subjective, time-consuming, and difficult to scale, motivating the adoption of computer vision techniques for more objective and consistent evaluation [13].

Early computer vision approaches for fruit quality assessment were based on handcrafted features such as color, texture, and shape descriptors. While these methods provided initial automation capabilities, they often lack robustness when dealing with variations in lighting conditions, occlusions, and natural biological diversity. In recent years, deep learning techniques particularly convolutional neural networks (CNNs) have significantly improved performance in image classification tasks by automatically learning hierarchical feature representations from raw data. This has made CNN-based approaches highly effective for distinguishing subtle variations in fruit quality [5, 8, 10, 11].

Despite these advances, many high-performing deep learning models are computationally intensive, limiting their applicability in real-time and embedded systems such as mobile robots, edge devices, and on-field sorting platforms. This constraint has driven the development of lightweight architectures that aim to balance accuracy with computational efficiency. Models such as MobileNetV3 are specifically designed for resource-constrained environments, offering reduced model size and faster inference while maintaining competitive accuracy [1].

In parallel, recent lightweight architectures such as YOLO26n have introduced highly optimized backbones tailored for real-time visual recognition tasks. The nano variant provides fast inference and reduced computational complexity, making it particularly suitable for embedded deployment. Its multi-scale feature extraction backbone enables rich representation learning while maintaining a minimal parameter count, making it a compelling candidate for fruit quality classification in resource-constrained environments [2, 3, 9].

In this context, this work investigates and compares different lightweight deep learning approaches for fruit quality classification, including a detection-based model (YOLO26n) [12], a mobile optimized classification network (MobileNetV3), and a custom-designed convolutional neural network implemented using TensorFlow. The objective is to evaluate the trade-offs between accuracy, computational complexity, and real-time performance across these different architectural paradigms.

The main contributions of this study are as follows:

1. The design and implementation of a compact CNN architecture with reduced parameter complexity;
2. A comparative analysis between detection-based and classification-based models for fruit quality inspection;
3. An evaluation of their suitability for embedded and real-time deployment, providing insights into their practical applicability in smart agriculture systems.

2. Methodology

2.1. Dataset and Preprocessing

The dataset employed in this study comprises of a total number equal to 12 977 images spanning eight classes before augmentation, corresponding to fresh and rotten samples of apples, bananas, lemons, and oranges. The dataset is partitioned into training, validation, and testing subsets with proportions of 87 %, 9 %, and 4 %, respectively. Following preprocessing and data augmentation, the total number of images increased to 30 373 while preserving the same split ratios, resulting in 26480 training images, 2596 validation images, and 1 297 testing images.

All images were preprocessed using automatic orientation correction and resized to a uniform resolution of 320 × 320 pixels to ensure consistency across inputs. To enhance model generalization and mitigate overfitting, several data augmentation techniques were applied during training, including horizontal and vertical flipping, as well as 90° rotations in both clockwise and counter-clockwise directions. Each original image generated three augmented samples, thereby increasing data variability and improving robustness to spatial transformations. Fig. 1 represents a look at some of the images present in this dataset.

Fig. 1. Confusion Matrix (MobileNetV3).

To provide a clearer overview of the dataset composition after preprocessing and augmentation, the distribution of images across all classes is illustrated in Fig. 2.

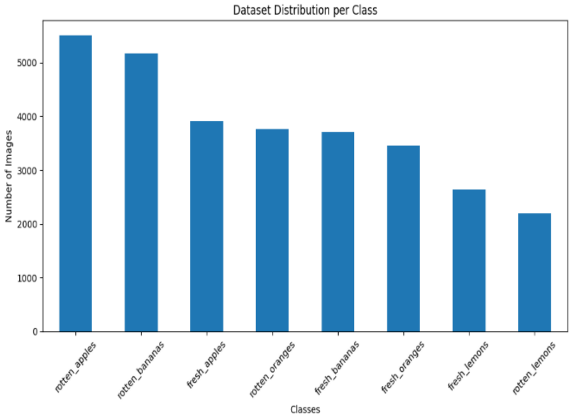


Fig. 2. Dataset Image Count Per Class.

As shown in Fig. 2, the dataset maintains a relatively balanced distribution among the eight classes, although slight variations in sample counts remain between categories. A detailed breakdown of the dataset partitioning into training, validation, and testing subsets for each class is presented in Table 1.

Table 1. Dataset Structure.

	train	Test	Valid	Total
Rotten_apples	4809	476	224	5509
Rotten_bananas	4500	444	235	5179
Fresh_apples	3379	351	184	3914
Rotten_oranges	3303	293	168	3764
Fresh_bananas	3260	320	135	3715
Fresh_oranges	3028	293	133	3454
Fresh_lemons	2301	225	112	2638
Rotten_lemons	1900	194	106	2200

Overall, the preprocessing and augmentation pipeline produced a comprehensive dataset suitable for fruit freshness classification.

2.2. Model Architecture

Three deep learning approaches are evaluated. First, a custom lightweight convolutional neural network (CNN) is designed with reduced depth and global average pooling (GAP) to minimize parameter count and computational cost. Second, a transfer learning approach based on a pretrained lightweight architecture is used, where the top layers are fine-tuned. Third, a YOLO-based model is trained to evaluate its architectures for classification tasks.

2.3. Mathematical Representation of the Proposed CNN Architecture

TensorFlow is an open-source deep learning framework developed by Google, widely used for building, training, and deploying machine learning models. It provides a flexible computational graph structure that supports both CPU and GPU acceleration. TensorFlow simplifies the implementation of complex neural networks through high-level APIs such as Keras. It is commonly used in computer vision, natural language processing, and large-scale AI systems [14, 15].

The proposed lightweight convolutional neural network is designed for efficient feature extraction and classification while maintaining low computational complexity for embedded deployment. Its core operations are as follow:

The equation below represents Feature Extraction Using Convolution Operation:

$$y(i,j) = \sum_m \sum_n x(i+m, j+n) \cdot w(m,n) \tag{1}$$

The Non-Linear Feature Activation with ReLU is presented via this function:

$$f(x) = max(0,x) \tag{2}$$

Spatial Feature Reduction is done through Global Average Pooling (GAP) provided by the equation below:

$$g_k = \left(\frac{1}{H \times W}\right) \sum_{i=1}^{H} \sum_{j=1}^{W} f_k(i,j) \tag{3}$$

The equation below presents the Probability Estimation Using Softmax Classification:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{4}$$

The Optimization Through Cross-Entropy Loss Minimization is provided by this equation below:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{5}$$

The Dimensionality Reduction via Max Pooling Operation is represented by the following equation:

$$y(i,j) = \max_{\{(m,n) \in R\}} x(i+m, j+n) \tag{6}$$

The Linear Transformation in the Fully Connected Layer can be expressed as:

$$yz = W\,x + b \tag{7}$$

The Regularization Using Dropout Mechanism is defined by the following expression:

$$\tilde{x} = x \cdot r, r \sim Bernoulli(p) \tag{8}$$

The Input Data Scaling and Normalization process is performed using the following equation:

$$x' = \frac{x}{255} \tag{9}$$

These operations collectively enable efficient learning, reduced parameter count, and strong generalization performance suitable for real-time applications.

2.4. Mathematical Representation of the Proposed MobileNetV3 Architecture

MobileNetV3 is a lightweight deep learning architecture designed for mobile and embedded devices. It improves computational efficiency using techniques such as depthwise separable convolutions and lightweight attention mechanisms, enabling fast and accurate image classification with low resource consumption. Compared to conventional CNNs, MobileNetV3 achieves a better balance between accuracy and computational cost, making it well suited for real-time applications in resource-constrained environments [4, 6, 7].

2.4.1. Depthwise Separable Convolution

Instead of standard convolution, MobileNetV3 decomposes it into two operations:

$$\begin{gathered} Y_{dw}(i,j,k) = \\ = \sum_{m,n} x(i+m, j+n, k) . w_{dw}(m,n,k), \end{gathered} \tag{10}$$

$$Y_{pw}\,(i,j,k) = \sum_k Y_{dw}(i,j,k) . w_{pw}(k,k'), \tag{11}$$

where Y_{dw} is the depthwise convolution output, Y_{pw} is the pointwise convolution output.

This factorization significantly reduces the number of parameters and operations compared to standard convolution.

2.4.2. Non-Linear Activation (Hard-Swish)

MobileNetV3 replaces ReLU with a more efficient activation function.

2.4.3. Batch Normalization

$$f(x) = \frac{ReLU6(x+3)}{6}, \tag{12}$$

where:

$$ReLU6(x) = min(max(0,x), 6) \tag{13}$$

2.4.4. Squeeze-and-Excitation (Channel Attention)

To enhance feature representation, MobileNetV3 applies channel-wise attention:

$$s = \sigma(W2 \cdot \delta(W1 \cdot z)), \tag{14}$$

$$\widetilde{F_k} = s_k \cdot F_k, \tag{15}$$

where $W1$ *and* $W2$ are the learnable weight matrices of the two fully connected layers inside the SE attention module, z is the globally pooled features (same GAP as in the proposed CNN), δ is the activation function, σ is the sigmoid function.

2.4.5. Final Classification

Similar to the proposed CNN, the extracted features are processed using global average pooling, followed by fully connected layers and Softmax classification. These operations are identical to Equations (3)–(5) and are therefore not repeated.

2.5. Mathematical Representation of YOLO26 Architecture

YOLO26n is a lightweight architecture belonging to the YOLO family, designed for efficient real-time visual recognition. Its nano-scale variant is optimized for minimal computational cost and fast inference, making it particularly suitable for embedded and resource-constrained applications. In this work, YOLO26n is employed as a classification model, leveraging its optimized multi-scale backbone for hierarchical feature extraction and its classification head for final category prediction.

Given an input image Img, the network extracts hierarchical feature representations and produces class scores through a final prediction layer.

$$F = \phi(Img), \tag{16}$$

where $\phi(\cdot)$ denotes the feature extraction network. Unlike the proposed CNN, which is specifically designed with reduced depth and handcrafted structure, YOLO26n utilizes a deeper and more optimized backbone originally intended for detection tasks, enabling richer feature representation.

The extracted feature maps are then transformed into class predictions through a classification head. Similar to the proposed CNN, global feature aggregation, fully connected mapping, Softmax activation, and cross-entropy loss are employed for final classification. Therefore, these operations are not repeated here.

The key distinction lies in the architecture design: YOLO26n benefits from a detection-oriented backbone that is inherently optimized for multi-scale feature extraction, which may enhance its ability to capture subtle variations in fruit quality compared to standard classification networks.

2.6. Training Strategy

All models were trained under strictly consistent conditions to ensure a fair and unbiased comparative evaluation. The training was conducted using Google Colab, with identical hyperparameters and data preprocessing pipelines applied across all architectures.

Training conditions and hyperparameters are presented in Table 2.

2.7. Models Evaluation

To evaluate the classification performance, we used standard metrics that capture both correctness and prediction confidence.

Table 2. Training conditions and hyperparameters.

Parameter	Value
Input resolution	320 × 320 pixel
Batch size	16
Maximum epochs	50
Optimizer	adam
Learning rate	0.001

Precision (P) measures the proportion of correctly predicted positive samples:

$$Precision\ (P) = \frac{TP}{(TP + FP)} \tag{17}$$

Recall (R) measures the proportion of actual positives correctly identified:

$$Recall\ (R) = \frac{TP}{(TP + FN)} \tag{18}$$

F1-Score balances precision and recall:

$$F1 - Score = 2 \times \frac{(P \times R)}{(P + R)} \tag{19}$$

Accuracy measures the overall correct predictions:

$$Accuracy\ \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{20}$$

Log Loss (cross-entropy loss) evaluates the difference between predicted probabilities and true labels:

$$log\ loss = -\left(\frac{1}{N}\right) \sum_{i=1}^{N} \sum_{j=1}^{C} y_{\mathrm{ij}} \log(p_{\mathrm{ij}}) \tag{21}$$

3. Results

A detailed evaluation of the proposed models is provided through confusion matrices and quantitative performance metrics. Figs. 3–5 illustrate the classification behavior of each model across all fruit categories, highlighting their ability to correctly distinguish between fresh and rotten samples. The confusion matrices provide a visual representation of prediction accuracy, misclassification patterns, and class-wise performance, thereby offering deeper insight into the strengths and limitations of each architecture.

Table 3 presents a detailed comparison of the evaluated models, combining both predictive performance metrics (accuracy, recall, F1-score, and log loss) and computational efficiency indicators (parameter count, inference time (on Nvdia GTX 1650

4GB), and model size), thereby highlighting the trade-offs between accuracy and deployment constraints.

All evaluated models achieved excellent classification performance with accuracies above 99.9 %, confirming the effectiveness of deep learning for fruit quality classification. Among them, YOLO26n achieved the best overall results, reaching 100 % accuracy with the lowest log loss, fastest inference time (28.41 ms), smallest model size (3.06 MB), but the highest parameter count (2.4M), making it highly suitable for real-time embedded applications for more powerful embedded hardware. MobileNetV3 and the proposed custom CNN also showed strong classification capability but MobileNetV3 required higher computational resources. Meanwhile the proposed custom CNN model had the lowest parameter count (0.4M). Although the results are highly promising, the high performance may be influenced by the relatively limited scope of the task and the small test dataset size. Therefore, further evaluation on larger and more diverse datasets is necessary to better assess model robustness and generalization.

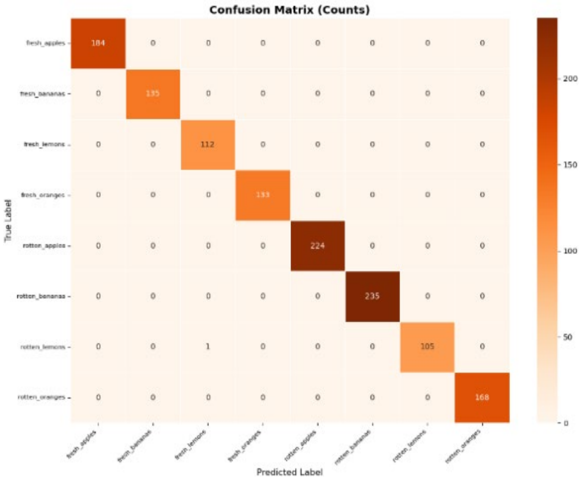


Fig. 3. Confusion Matrix (mobilenetV3).

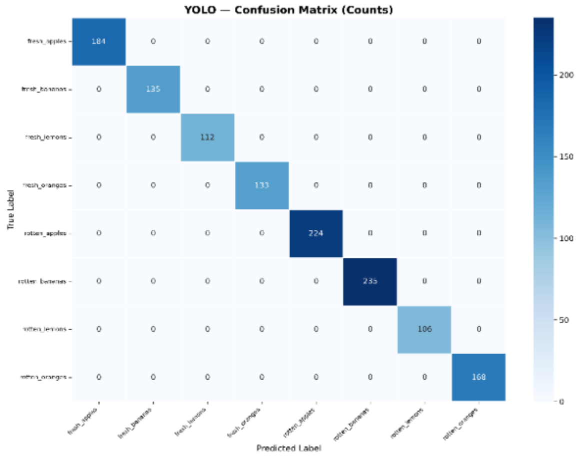


Fig. 4. Confusion Matrix (YOLO26n).

4. Conclusion

This study demonstrates that lightweight deep learning models can achieve excellent performance in fruit quality classification while satisfying the constraints of real-time and embedded deployment. Through a comparative analysis of a custom-designed CNN, MobileNetV3, and the detection-oriented YOLO26n architecture, the experimental results confirm that lightweight architectures are highly effective for smart agriculture applications. Among the evaluated models, YOLO26n achieved the best overall performance, reaching 100 % classification accuracy, a log loss of 0.0008, requiring a compact model size of 3.06 MB and the fastest inference time of 28.41 ms but it used 2.4 million parameters. These results highlight its strong suitability for real time edge and embedded systems. In comparison, MobileNetV3 and the proposed custom TensorFlow CNN both achieved 99.92 % accuracy, demonstrating competitive classification capability while maintaining relatively low computational complexity.

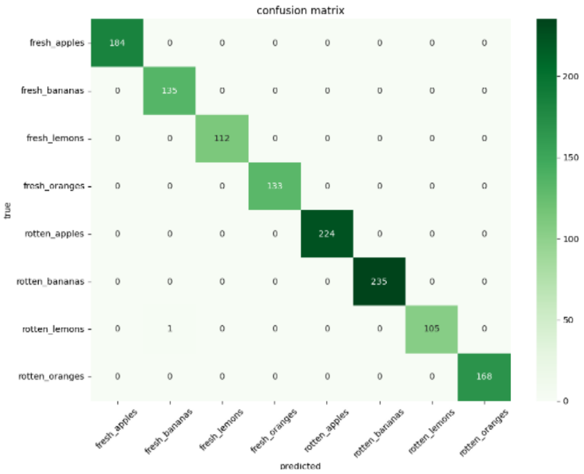


Fig. 5. Confusion Matrix (Custom TensorFlow).

Table 3. Models comparison.

	Yolo26n	Mobilenetv3	Custom TensorFlow
Test Accuracy	100 %	99.92 %	99.92 %
Recall	1	0.9992	0.9992
F1 - score	1	0.9992	0.9992
Log Loss	0.0008	0.0022	0.0130
Parameter number	2.4M	1.01M	0.42M
inference time (ms)	28.41	133.36	85.68
Model Size (MB)	3.06	14.6	8.13

The proposed custom CNN shows that optimized lightweight architectures can achieve strong generalization performance with a very low parameter count, without fully depending on large pretrained models. The evaluated models also highlight the potential of AI in automating agricultural inspection by reducing manual work, improving product consistency, and speeding up post-harvest quality assessment. Their low computational demands make them well suited for deployment on embedded and edge devices, enabling real-time decision-making directly in agricultural environments without relying on cloud processing.

Overall, the results show that lightweight deep learning methods are effective and efficient for automated fruit quality inspection in smart agriculture. Future improvements will include optimization techniques like quantization and pruning, along with deployment on embedded platforms such as Raspberry Pi 5 for real-time monitoring and robotic applications.

Acknowledgements

This work was supported by the EIPHI Graduate School (contract ANR-17-EURE-0002) and the Bourgogne-Franche-Comté Region).

References

[1]. Y. Gulzar, Fruit image classification model based on MobileNetV2 with deep transfer learning technique, *Sustainability*, Vol. 15, Issue 3, 2023, 1906.

[2]. B. Xiao, M. Nguyen, W. Q. Yan, Fruit ripeness identification using YOLOv8 model, *Multimedia Tools and Applications*, Vol. 83, Issue 9, 2024, pp. 28039-28056.

[3]. P. Kamat, S. Gite, H. Chandekar, L. Dlima, Multi-class fruit ripeness detection using YOLO and SSD object detection models, *Discover Applied Sciences*, Vol. 7, 2025, 931.

[4]. X. Yin, W. Li, Z. Li, L. Yi, Recognition of grape leaf diseases using MobileNetV3 and deep transfer learning, *International Journal of Agricultural and Biological Engineering*, Vol. 15, Issue 3, 2022, pp. 184-194.

[5]. H. S. Gill, G. Murugesan, A. Mehbodniya, G. S. Sajja, et al., Fruit type classification using deep learning and feature fusion, *Computers and Electronics in Agriculture*, Vol. 211, 2023, 107990.

[6]. Y. Zhang, S. Wang, G. Ji, Z. Dong, Fruit classification using computer vision and feedforward neural network, *Journal of Food Engineering*, Vol. 143, 2014, pp. 167-177.

[7]. M. H. Rahman, T. S. Akash, S. J. Fariha, Deep learning-based fruit recognition and quality assessment: A convolutional neural network approach, in *Proceedings of the International Conference on Smart Computing and Application (ICSCA)*, 2024, pp. 1-6.

[8]. M. Rizzo, M. Marcuzzo, A. Zangari, A. Gasparetto, et al., Fruit ripeness classification: A survey, *Artificial Intelligence in Agriculture*, Vol. 7, 2023, pp. 18-37.

[9]. L. E. Chuquimarca, B. X. Vintimilla, S. A. Velastin, A review of external quality inspection for fruit grading using computer vision and deep learning, *Artificial Intelligence in Agriculture*, Vol. 14, 2024, pp. 147-164.

[10]. B. Xiao, M. Nguyen, W. Q. Yan, Apple ripeness identification from digital images using transformers, *Multimedia Tools and Applications*, Vol. 83, Issue 3, 2024, pp. 7811-7825.

[11]. X. Zhu, F. Chen, Y. Zheng, C. Chen, et al., Detection of Camellia oleifera fruit maturity in orchards based on a modified lightweight YOLO network, *Computers and Electronics in Agriculture*, Vol. 226, 2024, 109471.

[12]. M. Chouhan, P. S. Banerjee, A. Kumar, FruitClass: A fruit classification system using modified MobilenetV3 with hard-swish function, *Applied Fruit Science*, Vol. 67, 2025, 47.

[13]. C. Guo, C. Zhu, Y. Liu, R. Huang, et al., End-to-end lightweight transformer-based neural network for grasp detection towards fruit robotic handling, *Computers and Electronics in Agriculture*, Vol. 221, 2024, 109014.

[14]. X. Deng, M. Shi, B. Khan, C. P. Lim, A lightweight CNN model for UAV-based image classification*, Soft Computing*, Vol. 29, Issue 4, 2025, pp. 2363-2378.

[15]. M. Afif, R. Ayachi, Y. Said, Deep embedded lightweight CNN network for indoor objects detection on FPGA, *Journal of Parallel and Distributed Computing*, Vol. 201, 2025, 105085.

(017)

Effect of Surface Curvature on CT-to-RGB-D Point Cloud Registration using Super4PCS and ICP Variants

R. Stilinović, M. Švaco, B. Šekoranja and F. Šuligoj
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lučića 5, 10000 Zagreb, Croatia
E-mail: roc.stilinovic@fsb.unizg.hr

Summary: This work investigates multimodal registration between computed tomography (CT)-derived and color-depth (RGB-D) point clouds for image-guided robotic procedures. The RGB-D point clouds are processed to retain different levels of surface curvature and selectively downsampled with the goal of reducing computational load while preserving registration accuracy. Registration is performed by combining the Super 4-Point Congruent Sets (Super4PCS) global method with Iterative Closest Point (ICP), weighted ICP (wICP), and Generalized ICP (GICP) for fine registration. The methods are evaluated across processing variants and camera viewpoints using one-sided trimmed Chamfer distance, trimmed Hausdorff distance, marker-based application accuracy (AA) errors, and runtime analysis. The results indicate that Super4PCS+GICP provides the best overall accuracy, although with the longest execution time, while comparisons between surface-processing variants show that preserved surface curvature influences registration quality and stability.

Keywords: Point cloud, Multimodal registration, Super4PCS, ICP, Curvature, Neurosurgical phantom.

---

1. Introduction

Robotic systems have been used in surgery since 1985, when the PUMA 200 robot was used to assist a surgical procedure [1]. Since then, they have become important tools for improving surgical precision and patient outcomes, although they are often associated with increased operative time [2]. Therefore, a key challenge in image-guided robotic surgery is to improve procedural efficiency while maintaining the accuracy required for safe intervention. This depends on accurate alignment between preoperative image data and intraoperative sensor data. In this work, we investigate multimodal CT-to-RGB-D point cloud registration using two head phantoms, focusing on how preserved surface curvature affects registration accuracy and execution time.

Image registration refers to computing correspondences and spatially aligning images [3], while multimodal registration applies this to data from different modalities, such as computed tomography (CT), ultrasound (US), magnetic resonance (MR), or color-depth (RGB-D) cameras. In robotic medicine, 3D multimodal registration enables anatomical information from different sources to be represented in a common spatial coordinate system. Recent studies have applied it in urology for magnetic resonance-transrectal ultrasound (MR-TRUS) image fusion [4], hepatobiliary surgery to align preoperative 3D liver models with intraoperative views [5], and neurosurgery to register the Cirq robotic system with cone-beam computed tomography (CB-CT) images fused with preoperative MR images [6]. Here, we focus on CT and RGB-D camera data in a neurosurgical application. The outer skin surface is reconstructed from CT DICOM data using the triplanar method [7], resulting in a 3D point cloud of the head surface. Registration is performed using the Super 4-Point Congruent Sets (Super4PCS) global method [8], followed by Iterative Closest Point (ICP) [9], weighted ICP (wICP) [10], and Generalized ICP (GICP) [11] for fine registration.

Before clinical implementation, the setup should be evaluated in a controlled environment that mimics the surgical workflow. Phantoms are suitable for this purpose because they represent target anatomy and selected functional properties. They have been used in robotic cardiology for patient-specific cardiac intervention experiments [12], abdominal surgery for preoperative puncture-trajectory testing [13], and neurosurgery for MRI-guided robotic stereotactic neurointervention [14]. In this work, two CT- and RGB-D-compatible head phantoms are used. Registration quality is evaluated using point-cloud distances, marker-based errors, and execution time to assess surface curvature preservation in terms of geometric accuracy and computational efficiency.

2. Materials and Methods

2.1. Pipeline

In this section we present the materials and methods used to conduct the experiments and evaluate the results. First, we show Fig. 1 containing the summary of our pipeline which is then described in detail in the following subsections.

2.2. Materials and Software

Two head phantoms were used: a non-openable phantom [15] and an openable phantom [16]. Before imaging, 3D-printed circular markers were glued to the back of the head, neck, and shoulder regions to enable later marker-based accuracy evaluation, and makeup was applied to reduce reflections, as shown in Fig. 2. Each phantom was first scanned with CT and then imaged at full resolution using a color-depth (RGB-D) camera [17] mounted on a robotic arm [18].

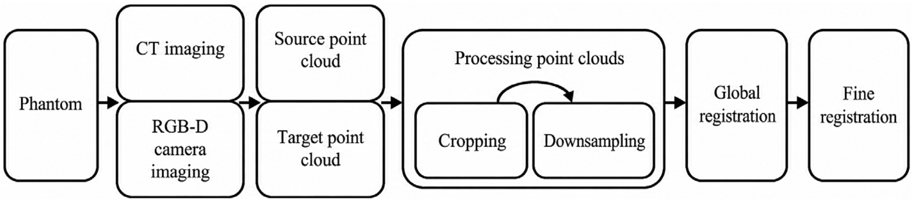


Fig. 1. Overview of the proposed pipeline. We start by placing the head phantom in the CT, take images, and convert them into point cloud using the method from [7] obtaining the source cloud for registration. Then the same phantom is placed on surgical table and imaged with an RGB-D camera mounted on the robot obtaining the target point cloud. Then the two point clouds are cropped (source cloud once to eliminate outliers, and target cloud multiple times to test the influence of curvature), downsampled using the selective method described in [20], and then registered; first globally with Super4PCS algorithm, and then locally using three ICP variations (ICP, GICP, wICP) for comparison.

Fig. 2. Head phantoms with circular markers:
(a) non-openable phantom with 21 markers
and (b) openable phantom with 15 markers. Both are shown in the head holder used for RGB-D imaging.

The implementation was carried out in C++ on a Linux-based workstation. Point cloud loading, processing, ICP, and GICP algorithms were performed using Point Cloud Library (PCL), while wICP was implemented manually as in [10]. Visualization toolkit (VTK) was used for rendering and for manual selection of point cloud regions that were cut or downsampled. The Super4PCS global registration algorithm was used through its open-source implementation [19], and a graphical user interface (GUI) was developed to integrate the visualization and registration components, as well as for testing the accuracy on marker points.

2.3. Setup

RGB-D acquisition was performed from three viewpoints while maintaining a fixed distance of 300 mm from the tip of the nose. The first viewpoint was acquired approximately along the surface normal determined in the slicer. The other two viewpoints were acquired by rotating the camera by +25° and −25° around the x-axis of the camera coordinate system, shown in Fig. 3, with respect to this initial direction.

For each RGB-D viewpoint, three cropped point-cloud datasets were generated to evaluate the influence of preserved surface curvature on registration. In the first dataset, only outliers such as the head holder and background were removed, while the visible head surface was preserved. In the second dataset, the neck and shoulder regions were additionally removed, leaving a reduced head-surface region. In the third dataset, the crop was further restricted by removing the forehead region, leaving mainly the facial surface. These three cropping variants are shown in Fig. 3 (a)–(c). Each cropped RGB-D point cloud was then downsampled using the selective method described in [20]. The cropping procedure is illustrated for the non-openable phantom, and the same procedure was applied to the openable phantom.

After that, we use the CT images to obtain 3D point clouds. The obtained point clouds are cropped so that the back of the head is removed, and approximately half of the head is kept for registration purposes, since the reduction of outliers may affect the registration accuracy. We show full and cropped CT images of non-openable phantom in Fig. 4.

2.4. Registration

After obtaining the CT-derived and RGB-D point clouds, registration was performed in two stages. First, the Super 4-Point Congruent Sets (Super4PCS) algorithm [8] was used for global registration to estimate an initial rigid transformation between the two point clouds. This step provides a coarse alignment suitable for partially overlapping point clouds. The obtained transformation was then refined using three fine-registration methods: Iterative Closest Point (ICP), weighted ICP (wICP), and Generalized ICP (GICP). These methods were selected to compare a standard point-to-point formulation, a residual-weighted formulation, and a formulation that also considers local surface structure.

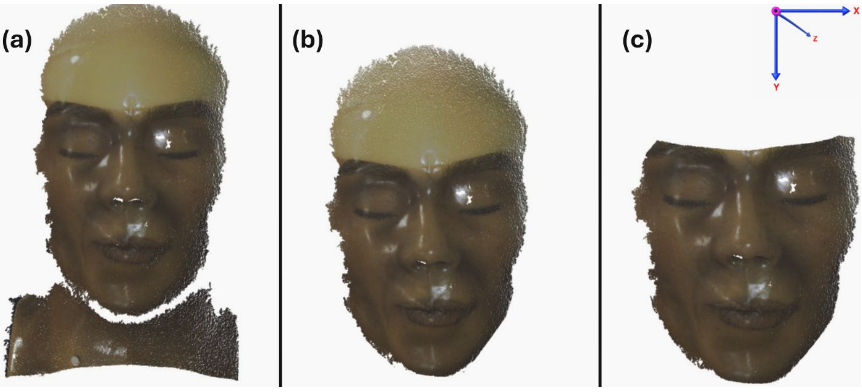


Fig. 3. The figure shows the three different datasets obtained from single view. Dataset (a) contains all the captured points that represent the head, and only the outliers like head holder or background are removed. In dataset (b) alongside outliers, also the neck and shoulder information is removed, while in dataset (c) alongside outliers, head and neck, also the forehead information was removed, leaving only facial points. Camera coordinate system can be seen in upper right corner.

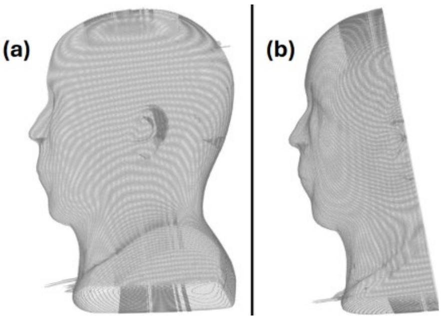


Fig. 4. Point clouds obtained from CT scans. In (a) we show the complete cloud, and in (b) the cropped part used for downsampling and registration.

The first variant is the classical ICP algorithm [9], which iteratively establishes correspondences between the point clouds and estimates the rigid transformation that minimizes the sum of squared Euclidean distances between corresponding points, as shown in Equation (1):

$$(R^{*}, t^{*}) = \arg\min_{R,t} \sum_{i=1}^{N} \| R\, p_i + t - q_i \|^2, \quad (1)$$

where $\boldsymbol{p_i}$ are points from the source cloud, $\boldsymbol{q_i}$ are the corresponding points from the target cloud, $\boldsymbol{R}$ is the rotation matrix, and $\boldsymbol{t}$ is the translation vector. Since all correspondence pairs contribute equally to the objective function, ICP is used as the baseline fine-registration method.

Unlike classical ICP, wICP assigns distance-based weights to the correspondences so that pairs with larger residuals contribute less to the optimization. This reduces the influence of poorly matched point pairs and outliers, which can appear more frequently when the available surface region is reduced by cropping. In this work, the weight is calculated as in Equation (2):

$$w_i = \begin{cases} 1, \|p_i - q_i\| < D \\ \frac{D}{\|p_i - q_i\|}, otherwise \end{cases}, \quad (2)$$

where $\boldsymbol{D}$ is median distance of all corresponding point pairs. The corresponding weighted cost function is given in Equation (3):

$$C = \frac{1}{m} \sum_{i=1}^{m} w_i \cdot \min_{j \in \{1,2,\ldots,n\}} \| p_i - q_j \|, \quad (3)$$

where $\boldsymbol{m}$ is the number of points from source point cloud. This formulation allows larger residuals to remain part of the optimization while reducing their effect on the estimated transformation.

The third method is GICP [11], which extends the point-to-point formulation by modelling the local surface structure around each point using covariance matrices. Instead of minimizing only Euclidean point distances, GICP minimizes cost function given in Equation (4):

$$E(R,t) = \sum_{i=1}^{N} r_i^{T} \left(C_i^{Q} + R\, C_i^{P} R^{T} \right)^{-1} r_i, \quad (4)$$

where $\boldsymbol{P}$ denotes source point cloud, $\boldsymbol{Q}$ target point cloud. $\boldsymbol{C_i}$ covariance matrix for each of the datasets, and $\boldsymbol{r_i}$ is the residual vector calculated as in Equation (5):

$$r_i = q_i - (R\, p_i + t) \quad (5)$$

Because GICP incorporates local surface geometry, it is expected to respond differently to changes in preserved surface curvature than ICP and wICP. Comparing these three fine-registration methods therefore allows the influence of the optimization criterion on registration accuracy and execution time to be evaluated under different cropping conditions.

2.5. Evaluation Metrics

Registration quality was evaluated using point-cloud distance measures, marker-based errors, and execution time. These measures were selected to assess both the geometric agreement between the registered point clouds and the practical computational cost of the registration pipeline.

For point-cloud distance evaluation, the nearest-neighbor distance from each point p_i in point cloud P to point cloud Q was first calculated as in Equation (6):

$$d_i = \min_{q_j \in Q} \left\| p_i - q_j \right\| \tag{6}$$

The resulting distances were then sorted in the ascending order as in Equation (7):

$$d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(m)}, \tag{7}$$

where m is the number of evaluated points. Since the registration problem involves partial-to-whole point-cloud alignment, a one-sided trimmed Chamfer distance was used. This metric is based on the Chamfer distance [21] and follows the trimmed formulation described in [20], where the largest 5 % of distances are treated as outliers and excluded. For a retained fraction $p = 0.95$, the one-sided trimmed Chamfer distance is calculated as in the Equation (8):

$$CD_p(P,Q) = \frac{1}{\lceil pm \rceil} \cdot \sum_{i=1}^{\lceil pm \rceil} d_{(i)} \tag{8}$$

To evaluate the largest remaining pointwise deviation after outlier rejection, the Hausdorff distance [22] was used in trimmed form, as shown in Equation (9):

$$H_p\,(P,Q) = d_{(\lceil pm \rceil)}, \tag{9}$$

where $\boldsymbol{p}$ is the retained fraction of points. In this work, the trimmed Chamfer distance represents the average registration residual after outlier removal, while the trimmed Hausdorff distance represents the worst retained residual.

Marker-based application accuracy (AA) was evaluated at selected physical locations on the phantoms. After registration, a robot-mounted laser [23] was repositioned to the transformed marker locations, and deviations from the physical marker centers were measured to include the complete chain from CT-to-RGB-D registration to robot-based navigation. The mean marker error was used as the main marker-based accuracy measure, while the maximum marker error was reported only as a supplementary worst-case indicator. Execution time was measured for the complete pipeline, including Super4PCS and the selected fine-registration method.

3. Results

The results were evaluated with respect to two main factors: the fine-registration method used after Super4PCS and the amount of surface geometry preserved in the RGB-D point cloud. The three fine-registration methods, ICP, wICP, and GICP, were applied to each cropping variant for both phantoms. In this way, the same registration pipeline was tested under different local refinement strategies and different levels of available surface information. The cropping variants were used to evaluate how performance changes when the RGB-D point cloud is reduced from a larger head surface to limited facial regions. For each crop, results were obtained from three RGB-D viewpoints to include the influence of camera pose.

The reported metrics include one-sided trimmed Chamfer distance, trimmed Hausdorff distance, mean marker-based application accuracy (AA mean), maximum marker error (AA max), and total registration time. The point-cloud distance metrics evaluate geometric agreement between the registered clouds, while marker-based AA evaluates alignment at selected physical locations on the phantoms. Distances and marker-based errors are expressed in mm, and execution time is expressed in s. All values except the maximum marker error are reported as averages over the three viewpoints. The maximum marker error represents the largest measured marker deviation for the corresponding crop and registration method and is reported as a supplementary worst-case indicator. Lower Chamfer, Hausdorff, and marker-based errors indicate better registration accuracy, while lower execution time indicates faster registration.

Across both phantoms, GICP achieved the lowest Chamfer and Hausdorff distances for all cropping variants, indicating the best overall point-cloud alignment. This suggests that including local surface structure in the fine-registration step improves the alignment compared with the point-to-point ICP formulation and the residual-weighted wICP formulation. The marker-based results generally follow the same trend, with GICP producing the lowest mean marker error in all tested cases. The only exception is the worst marker-based error for the full crop of the non-openable phantom, where GICP produced a larger maximum local deviation despite having the lowest mean marker error.

The influence of preserved surface curvature can be observed by comparing the full, forehead, and face crops. In general, the full crop provided the most stable results, especially in terms of marker-based mean error and execution time. This indicates that preserving a larger and geometrically richer surface region provides more reliable information for both the global Super4PCS alignment and the subsequent fine-registration step. However, the differences between crops are not uniform across all metrics. In some cases, especially for point-cloud distance measures, the forehead and face crops produced similar values. This suggests that registration quality is not determined only by the number of retained points, but also by the informativeness and reliability of the preserved surface geometry.

Table 1. Registration results for the non-openable phantom. Errors are in mm; time is in s.

Crop	Metric	ICP	wICP	GICP
full	Chamfer	0.513	0.511	0.445
	Hausdorff	1.107	1.106	0.924
	AA mean	1.681	1.674	1.416
	AA max	2.560	2.524	2.816
	Time	94.395	96.473	100.411
forehead	Chamfer	0.588	0.550	0.446
	Hausdorff	1.455	1.269	0.983
	AA mean	2.075	2.037	1.488
	AA max	2.713	2.624	2.250
	Time	166.852	168.361	170.963
face	Chamfer	0.543	0.528	0.452
	Hausdorff	1.227	1.177	0.987
	AA mean	1.775	1.716	1.515
	AA max	2.634	2.542	2.307
	Time	162.997	165.106	167.111

Table 2. Registration results for the openable phantom. Errors are in mm; time is in s.

Crop	Metric	ICP	wICP	GICP
full	Chamfer	0.358	0.339	0.307
	Hausdorff	0.782	0.729	0.626
	AA mean	1.231	1.256	1.069
	AA max	2.731	2.637	2.118
	Time	87.810	92.577	97.118
forehead	Chamfer	0.368	0.354	0.307
	Hausdorff	0.814	0.775	0.628
	AA mean	1.252	1.404	1.134
	AA max	2.579	2.728	1.983
	Time	129.495	135.736	142.792
face	Chamfer	0.387	0.388	0.305
	Hausdorff	0.879	0.878	0.625
	AA mean	1.449	1.657	1.160
	AA max	2.937	3.060	1.921
	Time	144.659	151.833	156.171

Execution time increased when the available surface region was reduced, particularly for the partial crops. GICP was consistently the slowest fine-registration method, which is expected because it uses local covariance information during optimization. Therefore, the results indicate a trade-off between accuracy and computation time: GICP provides the most accurate registration overall, while ICP and wICP are faster but more sensitive to reduced surface geometry.

4. Conclusions

The obtained results show that preserved surface curvature has a clear influence on registration quality, since the full datasets generally achieved lower marker-based errors and shorter registration times than the partial ones. This indicates that keeping a larger portion of the surface provides more stable geometric information for both the global Super4PCS step and the local refinement stage. Among the tested fine-registration methods, GICP achieved the best overall accuracy on both phantoms, particularly in terms of point-cloud distance measures and mean marker-based error. However, this improvement was obtained at the cost of longer execution time. ICP and wICP can therefore still be useful when faster execution is required, although their performance becomes more sensitive as the available surface geometry is reduced. The forehead crop occasionally performed worse than the face-only crop, especially for the non-openable phantom, which may be related to stronger reflection effects on the forehead surface.

Based on these experimental results, future work should explore patch-based registration as a possible way to retain the accuracy advantage of richer surface geometry while reducing data size and potentially improving global registration precision.

Acknowledgements

This research was funded by the project INSPIRATION – non-INvaSive PatIent RegistrATIon for rObotic Neurosurgery (grant no. NPOO.C3.2.R2-I1.06.0153), financed by the European Union through the National Recovery and Resilience Plan (NPOO).

We would also like to acknowledge the Regional Center of Excellence for Robotic Technology (CRTA) for their support of the project.

References

[1]. Á. Takács, D. Á. Nagy, I. J. Rudas, T. Haidegger, Origins of surgical robotics: From space to the operating room, *Acta Polytechnica Hungarica*, Vol. 13, Issue 1, 2016, pp. 13-30.

[2]. T.-J. Lai, C. Roxburgh, K. A. Boyd, J. Bouttell, Clinical effectiveness of robotic versus laparoscopic and open surgery: An overview of systematic reviews, *BMJ Open*, Vol. 14, Issue 9, 2024, e076750.

[3]. M. Chen, N. J. Tustison, R. Jena, J. C. Gee, Image registration: Fundamentals and recent advances based on deep learning, in Machine Learning for Brain Disorders (O. Colliot, Ed.), *Humana*, New York, 2023, pp. 435-458.

[4]. Z. Wang, M. Fan, Q. Tao, Q. Zhang, et al., Feasibility and safety study of advanced prostate biopsy robot

system based on MR-TRUS image flexible fusion technology in animal experiments, *SLAS Technology*, Vol. 29, Issue 5, 2024, 100184.

[5]. M. Y. Oh, K. C. Yoon, H.-J. Kong, T. Jang, et al., Leveraging augmented reality for dynamic guidance in 3-dimensional laparoscopic and robotic liver surgery: A prospective case series study, *Annals of Surgical Treatment and Research*, Vol. 109, Issue 1, 2025, pp. 44-52.

[6]. P. Truckenmueller, A. Früh, J. F. Kissner, N. K. Moser, et al., Integration of a lightweight and table-mounted robotic alignment tool with automated patient-to-image registration using robotic cone-beam CT for intracranial biopsies and stereotactic electroencephalography, *Neurosurgical Focus*, Vol. 57, Issue 6, 2024, E2.

[7]. J. Cvetić, B. Šekoranja, M. Švaco, F. Šuligoj, Triplanar point cloud reconstruction of head skin surface from computed tomography images in marker less image-guided surgery, *Bioengineering*, Vol. 12, Issue 5, 2025, 498.

[8]. N. Mellado, D. Auger, N. J. Mitra, Super 4PCS fast global point cloud registration via smart indexing, *Computer Graphics Forum*, Vol. 33, Issue 5, 2014, pp. 205-215.

[9]. P. J. Besl, N. D. McKay, A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, Issue 2, 1992, pp. 239-256.

[10]. Y. Su, Y. Sun, M. Hosny, W. Gao, et al., Facial landmark-guided surface matching for image-to-patient registration with an RGB-D camera, The International Journal of Medical Robotics and Computer Assisted Surgery, Vol. 18, Issue 3, 2022, e2373.

[11]. A. Segal, D. Hähnel, S. Thrun, Generalized-ICP, in *Proceedings of the Robotics: Science and Systems (RSS)*, 2009, 21.

[12]. Y. Wang, E. Yong, W. Gaozhang, V. Muthurangu, et al., MR-conditional robotic cardiac intervention: Design and validation in patient-specific phantoms, *IEEE Transactions on Medical Robotics and Bionics*, Vol. 7, Issue 4, 2025, pp. 1365-1376.

[13]. B. Zhang, K. Chen, Y. Yao, B. Wu, et al., Semi-automatic robotic puncture system based on deformable soft tissue point cloud registration, *International Journal of Computer Assisted Radiology and Surgery*, Vol. 20, Issue 3, 2025, pp. 525-534.

[14]. S. Huang, Z. He, Y. Chen, J. Chen, et al., Hybrid pneumatic-hydraulic actuation for MRI-guided robotic stereotactic neurointervention, *Science Advances*, Vol. 11, Issue 36, 2025, eady3624.

[15]. Adult head (static) for X-ray CT, US, MRI, True Phantom Solutions, https://www.truephantom.com/product/static-adult-head-x-ray-ct-us-mri/

[16]. Adult head (dynamic, advanced) for X-ray CT, US, MRI, True Phantom Solutions, https://www.truephantom.com/product/adult-head-dynamic-advanced-x-ray-ct-us-mri/

[17]. Ensenso B57-4 specification, IDS Imaging Development Systems GmbH, 2024, https://www.1stvision.com/cameras/IDS/Ensenso/dataman/Ensenso-B-series-datasheet.pdf

[18]. LBR iiwa 14 R820, KUKA, https://my.kuka.com/s/product/lbr-iiwa-14-r820/01t58000002hnktAAA?language = en_US&tab = Details

[19]. N. Mellado, D. Auger, Super4PCS, https://github.com/nmellado/Super4PCS

[20]. R. Stilinović, M. Švaco, B. Šekoranja, F. Šuligoj, Selective downsampling for fast and accurate 3D global registration with applications in medical imaging, *Mathematics*, Vol. 14, Issue 4, 2026, 606.

[21]. H. Fan, H. Su, L. J. Guibas, A point set generation network for 3D object reconstruction from a single image, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2463-2471.

[22]. D. P. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, Comparing images using the Hausdorff distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, Issue 9, 1993, pp. 850-863.

[23]. OM70-P0250.HH0130.VI, Baumer, https://www.baumer.com/int/en/product-overview/distance-measurement/laser-distance-sensors/high-performance-om70/high-measuring-accuracy/om70-p0250-hh0130-vi/p/38595

8th International Conference on Advances in Signal Processing and Artificial Intelligence
(ASPAI' 2026), 24-26 June 2026, Palma de Mallorca (Balearic Islands), Spain

(018)

Criteria for Integerization to Improve Matroid Computational Complexity

David W. Ash
Real Time Agents Inc, 191 Twinbridge Cir, Pleasant Hill, CA 94523-4739, USA
Tel.: +1 650 847 9893
E-mail: dash@alumni.stanford.edu

Summary: In recent years, algebraic matroids have emerged as a tool for analyzing dependencies in models in biological systems and signal processing. Understanding such dependencies can be useful in the design of experiments. Algorithms exist but are computationally expensive using known techniques such as Gröbner bases to find algebraic matroids. Techniques such as linearization and integerization have been explored for computing algebraic matroids in a less computationally expensive way. Such approaches have some theoretical limits, however. We establish criteria for the use of the less expensive techniques while staying within theoretical constraints, and argue that in practice such criteria will often set a relatively low bar.

Keywords: Algebraic matroids, Computational complexity, Linearization, Integerization, Biological models, Signal processing, Gröbner basis.

1. Introduction

A *matroid* is a combinatorial structure consisting of an ordered pair $(F, \mathcal{I})$ where F is a finite set and $\mathcal{I}$ is a collection of subsets of F. Although there are also infinite matroids, we will focus on finite ones. Each element of $\mathcal{I}$ is described as an *independent set*. Every subset of an independent set is also independent, and if A, B are independent sets with $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\}$ is independent. We refer to a maximal independent set as a *basis*. A particular type of matroid is an *algebraic matroid*. Within an ideal generated by a set of polynomials over the elements of a finite set of variables, certain sets of variables will or will not be algebraically independent. Considering the algebraically independent sets as the independent sets gives us a matroid, which we call an algebraic matroid. Not all matroids are isomorphic to an algebraic matroid ([7, 1]).

Computing algebraic matroids given the ideal can be computationally expensive. There are known techniques using Gröbner bases ([2, 4, 6]) for computing algebraic matroids, but these have exponential complexity. Rosen explored the concept of algebraic matroids in some detail, and showed that linearization is an approach that can reduce the need for expensive Gröbner basis calculations ([9, 10]). The author developed detailed algorithms which reduce, but do not completely eliminate, the need for the expensive Gröbner basis calculations. The author introduced an additional approach known as integerization. Essentially approaches such as linearization and integerization work by reducing the problem to one of performing computations in a simpler field [1].

It would therefore be desirable to completely eliminate the need for Gröbner bases calculations and work entirely using integerization in simpler fields. This abstract proposes a criterion for when we can do so.

2. Applications of Algebraic Matroids

Interest in algebraic matroids in recent years is partly driven by their potential application. Knowing which variables are, or are not, expected to be independent in any given situation is useful in the design of experiments ([9, 10]). This has, for example, been used to analyze the models of subsystems in the human body hosted at the BioModels website [5]. Algebraic matroids have also found application in signal processing [3].

3. Linearization and Integerization for Computing of Algebraic Matroids

Starting with earlier work by Ingleton [7], Rosen ([9, 10]) proposed the use of the linearization approach for computing algebraic matroids. By performing an algebraic transformation, it is possible to transform the problem of computing an algebraic matroid from a complex problem in a larger field to a simpler one involving linear calculations only over a smaller field. The author found that this approach does not completely eliminate the need for the complex Gröbner basis calculations, but does reduce their need. The author also introduced an additional simplifying technique, called integerization, allowing one to move the linearization calculations to the still simpler field of the rational numbers.

In more recent work, Rosen et. al. explore the possible limits of the linearization (and hence integerization) approaches. They caution that linearization works well in algebraically closed fields, but may not work so well if the field is not algebraically closed. Since they require that even the base field be algebraically closed, this is a criterion, but too restrictive to be desirable. We want to work over the rationals, which are not algebraically closed. We therefore propose a somewhat less restrictive criterion.

4. Criterion for the Use of Linearization and Integerization

Drawing on an earlier example of Pak [8], Rosen et.al. [11] cite an example of a matroid not conducive to linearization. They consider a field $K = \mathbb{Q}[t]/(t^2 + t - 1)$ and the matroid M which is the rank 3 linear matroid determined by the columns of the following matrix over K:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1+t \\ t & 0 & -1 & 1 & 0 & 1 & 1+t & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{1}$$

The reason why linearization fails here is that the equation $t^2 + t - 1 = 0$ has no solution in rationals. Of course, if we were operating over an algebraically closed field this would not be a problem. However, we propose the somewhat weaker criterion – rather than requiring that the field be algebraically closed we require only that **the set of polynomials generating the ideal have a non trivial solution in the rationals**. We can state rigorously that this criterion is necessary to use. In the example in Equation (1), columns 1, 5 and 9 are linearly dependent – hence not part of a basis – but this is only because $t^2 + t - 1 = 0$. Any replacement of t with a rational or integer will result in the three columns becoming linearly independent – and hence forming an extraneous basis.

Establishing whether the condition is also sufficient is more difficult rigorously. However, by contrast with the example given above by Rosen et.al., other matroids arising in, for example, biomedical applications do have this property. Consider, for example, the Schmitz model from [10]. The ideal in this case is generated by the following polynomials:

$$\begin{gathered}
\delta_0 - \delta_1 X + \delta_2 X_n - \delta_5 X Y_a + \delta_6 C_{XY} = 0, \\
-\delta_3 Y_a + \delta_4 Y_{an} - \delta_5 X Y_a + (\delta_6 + \delta_7) C_{XY} - \\
-\delta_{15} Y_a + \delta_{16} Y_i = 0, \\
\delta_1 X - \delta_2 X_n - \delta_8 X_n Y_{an} + \delta_9 C_{XYn} - \\
-\delta_{11} X_n T + \delta_{12} C_{XT} = 0, \\
\delta_3 Y_a - \delta_4 Y_{an} - \delta_8 X_n Y_{an} + \\
+(\delta_9 + \delta_{10}) C_{XYn} = 0, \\
\delta_5 X Y_a - (\delta_6 + \delta_7) C_{XY} = 0, \\
\delta_8 X_n Y_{an} - (\delta_9 + \delta_{10}) C_{XYn} = 0, \\
-\delta_{11} X_n T + \delta_{12} C_{XT} = 0, \\
\delta_{11} X_n T - \delta_{12} C_{XT} = 0, \\
\delta_7 C_{XY} - \delta_{13} X_p = 0, \\
\delta_{10} C_{XYn} - \delta_{14} X_{pn} = 0, \\
\delta_{15} Y_a - \delta_{16} Y_i = 0
\end{gathered} \tag{2}$$

It is fairly easily shown that this set of equations does have a non-trivial solution in the rationals. Moreover, as shown in [9, 10] and [1] this matroid is amenable to linearization and integerization approaches. The reason why the criterion makes sense can be seen from the example given above from [11]. With the matrix in (1), the first, fifth, and ninth columns are linearly dependent. However, there is no rational or integer assignment of a value to t that can capture this dependence, making a linearization approach infeasible. This problem does not arise with the system in (2).

5. Future Work

Linearization and integerization as approaches for computing algebraic matroids have drawn some attention in the literature in recent years. The question of how generally applicable these approaches are having therefore been raised. The current work has proposed a somewhat more general criterion for when these approaches can be used. This does suggest two important areas for future work.

First, it is conjectured – but not yet shown – that for most matroids arising in practice, the resemblance will be more to (2) where there is a possible solution in rational numbers, and less to (1). Verifying this through the analysis of a large number of models from the BioModels website, or elsewhere, would be very helpful in validating this approach.

Second, the theoretical results obtained by Rosen et.al. in [11] should be extended so that there is a theoretical underpinning for the use of linearization and integerization for a wider range of matroids, rather than just those over algebraically closed fields.

In particular, it would be conjectured – but not yet proven – that a refinement of the integerization approach would be helpful here. Whereas integerization as previously used simply assigns integers to variables randomly, it would be proposed instead to assign rational numbers to variables based on solutions to the set of equations generating the ideal. Two theoretical questions then arise. One is whether it is sufficient to do the linearization analysis using a particular such assignment, or whether in general all such assignments would need to be tested. The second is the weaker question of whether, if linearization shows no dependency for any such assignments, we can assume that none exists.

References

[1]. D. W. Ash, Linearization and integerization for computing algebraic matroids, in Advances in Signal Processing, Vol. 3, *IFSA Publishing*, Barcelona, 2025, pp. 23-38.

[2]. T. Becker, V. Weispfenning, Gröbner Bases: A Computational Approach to Commutative Algebra, *Springer*, New York, 1993.

[3]. D. I. Bernstein, C. Farnsworth, J. I. Rodriguez, The algebraic matroid of the finite unit norm tight frame (FUNTF) variety, *Journal of Pure and Applied Algebra*, Vol. 224, Issue 8, 2020, 106351.

[4]. B. Buchberger, An algorithm for finding the basis elements of the residue class ring modulo a zero-dimensional polynomial ideal, PhD Thesis, *Universität Innsbruck*, Innsbruck, 1966 (in German).

[5]. BioModels database, EMBL-EBI, 2006, https://www.ebi.ac.uk/biomodels

[6]. M. Gallinger, Gröbner bases: Ideal membership and graph colouring, Honours Thesis, *Lakehead University*, Thunder Bay, 2013.

[7]. A. W. Ingleton, Representation of matroids, in Combinatorial Mathematics and Its Applications (D. J. A. Welsh, Ed.), *Academic Press*, London, 1971, pp. 149-167.

[8]. I. Pak, Lectures on Discrete and Polyhedral Geometry, 2010, http://www.math.ucla.edu/~pak/book.htm

[9]. Z. Rosen, Algebraic matroids in applications, PhD Thesis, *University of California*, Berkeley, Berkeley, 2015.

[10]. Z. Rosen, Supplemental material (matroids.m2) for: Algebraic matroids in applications, 2015, https://github.com/zvihr/algebraic-matroids/blob/master/matroids.m2

[11]. Z. Rosen, J. Sidman, L. Theran, Linearizing algebraic matroids, *arXiv*, 2025, arXiv:2507.07220.

(023)

Applied Pseudo-Expert System Approach to Load-Bearing Performance of Mass Timber Panels Including Non-Rigid Bonding and Ply Angle

R. Hussein
State University of New York (emeritus), Syracuse, NY 13210, USA
Tel.: + 001 6802870342
E-mail: ezpscg@gmail.com

Summary: Literature lacks studies on cross-laminated timber (CLT) and diagonal-laminated timber (DLT) that consider essential properties such as interlayer bonding, the ply angle, and applied design apps. This paper addresses these issues in three parts. The first part aims at the bonding issue. A rigorous analytic model was derived. Real physical properties were considered in the formulation, which has been analyzed, discussed, and validated. The findings show that bonding stiffness must not be undervalued in serviceability, and the CLT is better suited for panels subjected to transverse loads than DLT. Secondly, a new practical formula is introduced. Thirdly, a fresh pseudo-expert system (-ES) has been coined. It introduces novelty and innovation through its hybrid conceptualization of the design process. The paper includes two applications (apps). The first is for CLT and DLT. The second is called Solver. Both apps are coded using Python, MATLAB, and Visual Studio.

Keywords: CLT, DLT, Non-rigid bonding, Ply angle, Pseudo expert system, Series-type solutions, Engineering software.

---

1. Introduction

Engineering design formulas for the CLT calculations are incorporated into international design criteria, such as the American, Canadian, European Union, Italian, Finnish, Swedish, British, and German standards. However, there are currently no comparable specifications for the DLT, though it's popular nowadays in various parts of the world.

From an engineering design perspective about laminates, literature relies on the assumption of perfectly rigid bonding. That assumption is unrealistic. In this paper, we examined the degree of actual bond rigidity and its effect on the structural performance of CLT and DLT panels, such as deformations.

For interlayer flexibility, which affects structural performance, some websites suggested integrating the Hankinson formula together with the shear analogy technique for calculating the flexural stiffness of the DLT. First, this integration is rigorously baseless, and so, irrelevant to structural invention [1]. The Hankinson formula was empirically introduced from the idea of in-plane failure theory rather than the rigorous out-of-plane behavior.

Secondly, Arnold applied Hankinson's formula for the off-axis modulus of elasticity transformation in his doctoral thesis [2], which he then used for in-plane stress analysis only. Applying his idea to DLT is also unfounded because it omits any references that link the in-plane behavior of the Hankinson formula to the Bernoulli-Euler theory, which emphasizes flexural behavior.

Thirdly, a typical CLT has a length-to-width ratio of 6 to 30, according to the American Plywood Association [3]. This range of ratios demonstrates how well CLT panels perform in engineering mechanics calculations as beams under flexure. Additionally, DLT has been studied in literature only under in-plane loads [2, 4, 5].

Those aspects of existing literature have been revisited in this paper. The paper considers the real physical properties of all materials in engineering calculations [6, 7] and assesses their impacts. Since there are no pertinent closed-form solutions, all the governing equations and boundary conditions are developed, solved, and satisfied using Fourier trigonometric series. Different loading scenarios and boundary circumstances can be accommodated by series-type solutions [8-10]. Our series-type modeling and solution was applied extensively in the literature for various types of engineered laminates and loads [11-17] and acknowledged by NASA [6]. The validity and accuracy of our methodological approach is ascertained by comparing the computed deflections, as an example, with the analytical results published in the literature in reputable sources. All the results concurred satisfactorily. The effects of interlayer deformation and ply angles on the structural performance are analyzed and discussed.

This paper is ongoing research and extends the scope of our previous studies and the state of the art of CLT and DLT panels. It introduces and demonstrates novelty and innovation through its fresh conceptualization and adaptable application of the design process and architecture.

In general, expert systems either codify human expertise into a knowledge-based or integrate the computational framework and basis for design with the engineer designated as the natural interface component or the interaction bridge for adaptive decision-making capabilities. We chose the latter strategy because the human interface is the one who operates the system, enters input-data, and interprets, validates, and refines its output. This way is imperative because humans can understand ambiguous situations better than predefined rules and can think outside the system.

In addition, the structural design is neither purely mathematic nor systematically linearly processed, and it encompasses immeasurable elements. The direct involvement of the design engineers in the process was always an inseparable component of it. This pathway is domain-specific, i.e., a specific specialized area, because each design problem is unique and what separates our concept from typical AI methods, which are broadly applicable across multiple domains. It introduces a hybrid ES where the algorithm handles computations, and the human guides the system, interprets the results, and handles the decisions and recommendations.

Technically, the fresh approach is a computer program that stores and applies rules, constraints, and decision criteria in if-then computable format and requires the designer as the interface to interact and control it manually. Unlike AI methods, the system's procedural operation, inputs data, and priori unknown outputs apply human judgment for logical reasoning to solve problems or recommend actions for compliance with well-defined domain specific areas, such as specific standard design requirements. Literature has references, e.g., [18], that sufficiently elaborate on expert systems and provide examples of pseudocodes and snippets. The pseudo-ES is not inherently about repackaging tasks; i.e., it does not use a known method without fundamentally changing the rules, reshuffling, reformatting, or reorganizing the existing engineering solutions.

2. Part I: Analytical Modeling and Solutions

For our analytical model and solutions, consider a panel of length and width L and b and depth d1 and d2. The subscript 1 indicates the top layer. The panel is under a uniform transverse load of intensity q_o, N/m. Interlayer flexibility could be due to bonding or roller shear in the transverse layers. They should not be confused. The bonding is responsible for joining the layers together. Adhesive shear is an interlaminar mechanical stress, whereas rolling shear is an inherent material property. This section aims at the effects of bonding.

The derivation begins with the following assumptions [6, 7, 9, 22]:

1. The material's behavior is linearly elastic;
2. The panel analysis is based on small deflection theory;
3. The panel is composed of two layers, each with a uniform elastic modulus (Ei), constant moment of inertia (Ii), m^4, and cross-sectional area (Ai), m^2, for layer i;
4. The panel is symmetrical about its centroidal vertical axis;
5. Each layer obeys Bernoulli-Euler beam theory for bending;
6. The connection between the layers is characterized as a continuous shear connection with a constant linear elastic shear slip modulus;
7. The panel is under bending loads only.

If the layers have the same curvature, then [8]:

$$\frac{M_1}{E_1 I_1} = \frac{M_2}{E_2 I_2} = -\frac{d^2 w}{d x^2}, \tag{1}$$

where for layer i, Ei is the modulus of elasticity, Pa, Ii is the cross-sectional moment of inertia about the layer neural axis, m^4, M_i is the internal bending moment around the layer neutral axis, N.m., w is the deflection, m, and x is the panel coordinate axis, m. with its origin at the left end.

Equation (1) can be rewritten as follows [9, 10, 20, 26]:

$$M_1 + M_2 = -EI \frac{d^2 w}{d x^2}, \tag{2}$$

where EI is the panel's flexural stiffness. Since the panel is not subjected to any in-plane loads, then

$$\sum_{i=1}^{2} N_i = 0, \tag{3}$$

where Ni = the layer internal axial force, N. At the interface between the layers, the compatibility condition is expressed as [6]:

$$V = k\,du = k\,(u_1 \cos^2\theta - u_2), \tag{4}$$

where k is the bonding coupling or connecting shear stiffness per unit length, which according to Eq. (4) must be in units of stress, Pa, θ is the ply-angle in radians, ui is the in-plane interlayer deformation, m, and V is the interlayer shear flux or flow, N/m. It should be noted that the ply angle relates to the individual layers. The engineer selects the angles that leverage the structural performance of the panel at hand.

According to the top layer's equilibrium equation

$$E_1 A_1 \cos^2\theta \frac{d^2 u_1}{dx^2} + V = 0 \tag{5}$$

There must be an equivalent relationship between the applied bending moment, Ma, N.m. and the resisting moment, Mr, N.m. Thus [26]

$$M_a = -EI \frac{d^2 w}{d x^2} - E_1 A_1 \cos^2\theta \, d \, \frac{du_1}{dx}, \tag{6}$$

where d is the distance from layers center to center, m, and EI is expressed according to the fundamentals of engineering mechanics [9, 20] as follows

$$EI = E_1 I_1 \cos^2\theta - E_2 I_2$$

The load-deflection relationship is given by [8-10, 20, 25]

$$\frac{d^2 M}{d x^2} = EI \frac{d^4 w}{d x^4} - E_1 A_1 \, d \cos^2\theta \frac{d^3 u_1}{d x^3} = q \tag{7}$$

The solution to Eq. (7) must satisfy the boundary conditions, which for simply supported panels are w = 0 at x = 0 and x = L, N = 0 at x and x = L.

For solving the previous equations, the displacements and applied load are represented as trigonometric series as follows:

$$w(x) = \sum_{i=1,2,..}^{\infty} W_i \sin \alpha_I x, \tag{8}$$

$$u_1(x) = \sum_{i=1,2,..}^{\infty} u_{1i} \cos \alpha_I x, \tag{9}$$

$$u_2(x) = \sum_{i=1,2,..}^{\infty} u_{2i} \cos \alpha_I x, \tag{10}$$

$$q = \sum_{i=1,2,..}^{\infty} q_i \sin \alpha_I x, \tag{11}$$

in which $\alpha_i = \frac{i\,\pi}{L}$, u_{1i}, u_{2i}, and W_i are coefficients, $q_i = \frac{4\,q_o}{i\,\pi}$ for a uniformly applied load of intensity, Pa. The three unknowns so far can now be determined as follows. By substituting Eqs. (9) and (10) in Eq. (3), the first determining formula is obtained as follows:

$$E_1\,A_1(u_{1i}\cos^2\theta + u_{2i}) = 0 \tag{12}$$

Also, by substituting Eqs. (8), (9), and (10) in Eq. (5) yields

$$k\,d\,\alpha_i W_i - (E_1\,A_1 u_{1i}\cos^2\theta\,\alpha_i^2 + k)\,u_{1i} + k\,u_{2i} = 0 \tag{13}$$

Combining Equations (12) and (13) yields the following second equation:

$$\frac{k\,d\,\alpha_i W_i}{E_1\,A_1} - \left(\alpha_i^2 + \frac{k\cos^2\theta}{EA}\right) u_{1i} = 0, \tag{14}$$

in which $EA = \frac{E_1\,A_1\,E_2\,A_2\,\cos^2\theta}{E_1\,A_1\cos^2\theta - E_2\,A_2}$.

Eq. (7) is now combined with Eqs. (11), (8), and (9) to yield the third governing formula.

$$EI\,\alpha_i^4\,W_i + E_1\,A_1\,d\,\cos^2\theta\,\alpha_i^3\,u_{1i} = q_i \tag{15}$$

Finally, Wi is determined by solving Eqs. (14) and (15)

$$W_i = \frac{q_i}{\alpha_i^4\,(EI - \frac{k\,d^2}{\alpha_i^2 + \frac{k}{EA}})} \tag{16}$$

For a three-layer panel under uniform load, the following solution is obtained by applying the previous formulation of two layers:

$$W_i = \frac{4\,q_o\,L^2}{(\pi\,i)^5\,EI}\,[\frac{2\,d^2}{\left(2\,d^2 - \frac{EI}{E_1\,A_1\cos^2\theta} + \frac{1}{k}\left(\frac{I\,\pi}{L}\right)^2\right)} - 1], \tag{17}$$

where d is the distance between the mid-planes of adjacent layers, and 1, 2, and 3 stand for the top, middle, and bottom layers, respectively.

3. Validation and Parametric Analysis

Literature lacks rigorous analytic formulation and solutions for the performance of CLT and DLT panels. The accuracy of the above-described model and solutions was validated using a selection of examples with answers from credible sources in the literature [9, 20, and 21]. Various two- and three-layer panels with the following characteristics were considered in the validation: a uniform load of q = 160 N/m, width = 0.24 m, and thickness = 0.3 m. Convergent solutions with satisfactory accuracy can be obtained with a finite number of terms. In our validation, a fast convergence can be observed using 100 terms. Table 1 compares the results for two- and three-layer rigidly connected simple panels.

Table 1. Comparison of the maximum deflection.

Number of Layers	L (m)	E (GPa)	Present Solution (mm)	Ref. [9, 19] (mm)	Col. 4 ÷ Col.5 (%)
2	3.66	68.9	0.001256	0.001256	100
2	10.0	68.9	0.069993	0.069993	100
2	3.66	65.0	0.001331	0.001331	100
2	3.66	75.0	0.001154	0.001154	100
3	3.66	68.9	0.000372	0.000372	100
3	20.0	68.9	0.331820	0.331820	100
3	3.66	65.0	0.000394	0.000394	100
3	3.66	75.0	0.000342	0.000342	100

Table 1 indicates the satisfactory accuracy of our exact closed-form rigorous solutions using the percentage of the results from the series-type solution to the well-established and applied-in-literature fundamental classic values [9, 19, 20]. Therefore, it would not be an apple-to-apple comparison to use numerical and approximate methods like finite element and experimental data. Table 1, together with the additional nine cases from Table 2, representing the maximum practical number of layers, suggests confidence in our modeling and solutions. It should be noted that increasing the number of layers will not change the accuracy of the comparison outcome.

This discovery highlights a crucial design element. Bonding and adhering layers should not be considered individually in structural design, as we found in our earlier laminate research [11-17]. Together, they both affect the performance. Weak bonding materials result in significant interlayer deformations, which affects the panel overall. However, weak adhering layers wouldn't need a very stiff bond and vice versa would be unwise economically. The bonding may be considered fully rigid once its stiffness reaches a point where the panel performance change can be essentially ignored. This finding leverages the economic aspect of the panel's design. It should be noted that Figs. 1 and 2 are not reported in the literature for CLT and DLT.

4. Results, Analysis, and Discussion

Analyzing and discussing results obtained from solving realistic panels is crucial to understand and confidently designing the load-bearing CLT and DLT while accounting for the necessary yet have been overlooked in the literature.

To this end, the data from the previous section is used. What constitutes the ubiquitous perfectly rigid bonding for laminates; quantifying what is regarded as assumptions in the literature; how the choice of constituent species affects the economical design of the panels; whether there is a boundary between CLT and DLT where one becomes preferred over the other for applications are just a few of the crucial issues to address and performance aspects to explore. It is worth noting that the structural design is a vast area, and providing the right analytical tools should empower designers to address unforeseeable challenges.

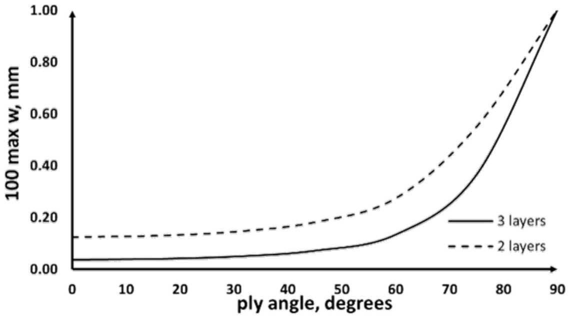


Fig. 1. Effect of ply-angle on deflection.

Fig. 1 shows the effects of ply angle on the maximum deflection of two- and three-layer panels. The deflection increases by 14 % when the angle is increased from 10 degrees by two orders of magnitude and by 164 % when the angle is increased from 60 degrees by the same order. This indicates that the larger the ply angle, the less effective the transverse layers and therefore the larger the deflection.

In addition to engineering commonsense, this outcome confirms that CLT is more appropriate than DLT for flexural load applications. Diagonalization of layers is helpful for in-plane loads because of the fundamentals of mechanics [9, 20]. At an angle of about 45°, the deflection starts to sharply increase. This angle could be considered as the boundary between CLT and DLT behaviors.

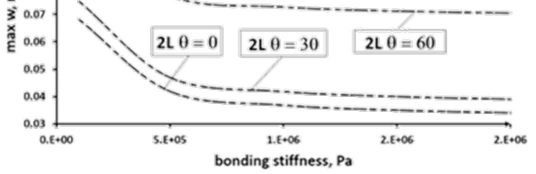


Fig. 2. Effect of the bonding stiffness on deflection.

Several published studies established load-slip curves for wood connectors from which the bonding stiffness is determined [19, 27]. Those findings go counter to the popular assumption of perfectly rigid bonding. The effects of bonding stiffness and ply-angle on the maximum deflections of two-layer panels are depicted in Fig. 2. For all ply angles, the deflection is more sensitive to variations in the lower range of the bonding stiffness value.

This discovery highlights a crucial design element. Bonding and adhering layers should not be considered individually in structural design, as we found in our earlier laminate research [11-17]. Together, they both affect the performance. Weak bonding materials result in significant interlayer deformations, which affects the panel overall performance. However, weak adhering layers wouldn't need a very stiff bond and vice versa would be unwise economically. The bonding may be considered fully rigid once its stiffness reaches a point where the panel performance change can be essentially ignored. This finding leverages the economic aspect of the panel's design. It should be noted that Figs. 1 and 2 are not reported in the literature for CLT and DLT.

5. Part II: Practical Analytical Method for Bending Stiffness Including Bonding and Ply Angle

Practical analytical methods are essential for a variety of reasons, including:

1. Filling in the gaps between theoretical mathematics and useful real-world design situations;
2. Provide data and insights into the interactions between various design variables to make well-informed, sound engineering decisions;
3. Getting results fast without sacrificing accuracy;
4. Reduce the possibility of human error in comparison to computations done by hand.

The bending stiffness of CLT panels can be determined using formulas provided by the United States, Canada, the European Committee for Standardization, the Engineered Wood Association, the Italian National Research Council, the Italian Standardization Organization, the Finnish Standards Association, the Swiss Association for Standards, the National Standards Body of the United Kingdom, and the German National Standardization Body [20-24]. Though well respected, some of the shortcomings of those formulas include lack of bonding stiffness and the ply angle, which are essential for everyday customary design. This section fills this gap via our pertinent completed studies [11-17].

The CLT panels are idealized in literature as two components representing the flexural stiffness of individual layers and shear stiffness of connections. This conceptualization is used under different titles such as "gamma," "shear analogy," and "Timoshenko" theory methods [22]. The effective bending stiffness, or EIeff, for the structural panels with fully rigid bonding refers to the panels' resistance to flexural deformation under transverse loads. It is an intrinsic fundamental property, influenced by the modulus of elasticity and the moment of inertia of the cross-section, and is defined by the classical Euler-Bernoulli theory as follows [7, 9, 21, 22]:

$$(\mathrm{EI})_{\mathrm{eff}} = \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{EI})_{\mathrm{i}}, \tag{18}$$

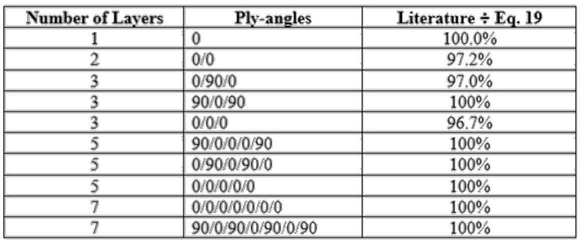
where i is the i^{th} layer moment of inertia and E is the modulus of elasticity of its material. The cross layers are homogenized into equivalent continuous shearing coupling. This accommodates the rolling shear, where the shear deformations in the longitudinal direction are ignored. The apparent bending stiffness that accounts for the shear deformation and is given by [11-17, 22]:

$$(\mathrm{EI})_{\mathrm{app}} = \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{EI}\,(\cos\theta)^2)_{\mathrm{i}} + \frac{(\mathrm{E\,A\,a^2})_{\mathrm{i}}}{\left[1+\frac{\pi^2\,\mathrm{E\,A}}{\mathrm{k\,L^2}}\right]_{\mathrm{i}}}, \tag{19}$$

where L is the panel's span; k is the real bonding stiffness, Pa; a is the distance of the ith layer-to-panel centroids; A is the cross-sectional area, m^2, and θ is the ply angle. Equation (19) is referred to as Hussein's formula, just like Hankinson's formula. Both formulas are intended for real-world applications and are simple to apply in many practical design scenarios.

Equation (19) is validated using sufficient practical configurations of CLT and DLT from one to seven rigidly connected layers and the following parameters: $E = 69$ GPa, $L = 3.66$ m, $A = 0.015$ m^2, and $I = 3.33 \times 10^{-6}$ m^4. Table 2 shows the results.

Table 2. Comparison of the bending stiffness using Eq. (19) and literature [7, 9, 21].



Number of Layers	Ply-angles	Literature ÷ Eq. 19
1	0	100.0%
2	0/0	97.2%
3	0/90/0	97.0%
3	90/0/90	100%
3	0/0/0	96.7%
5	90/0/0/0/90	100%
5	0/90/0/90/0	100%
5	0/0/0/0/0	100%
7	0/0/0/0/0/0/0	100%
7	90/0/90/0/90/0/90	100%

6. Part III: Interactive Apps for CLT and DLT

Practical digital apps are useful to programmatically automate the manual, time-consuming and cumbersome calculations, and eliminate the potential errors in the iterative design processes. They incorporate advantages including, but not limited to, the following:

- Achieving design specifications such as stiffness by varying layer stacking sequences, the ply orientations, and using hybrid species;
- Meeting design or practical limits, e.g., standard allowable deformations, manufacturing and assembling ply angles, available thicknesses in local markets, mechanical properties of available species, etc.;
- Providing designers with tools to test various design scenarios and variables by adjusting a set of variables within predefined constraints, etc.

In our search for pertinent digital tools that adopt classic and overlooked or ignored design variables, we found that pseudo-ES appealing. In addition to the advantages mentioned, they conceptualize the two necessary components of any structural design: namely, technical aspects and live human engagement. The app implements an algorithm for the presented model and solution, whereas the second one implements another algorithm to determine an optimum solution to a defined objective function; its end product is named a "solver" because of the widespread use of this term in commercial products.

The structural design process, including that for CLT and DLT, is a cyclical process that requires continually refining a design from concept through construction. The ethos behind this iterative process is to improve the design with each iteration until all requirements are met. This approach requires computational tools because the process is not systematic, time-consuming, and cumbersome. Therefore, the two distinct apps complement each other and are necessary for real-world applications because the process is neither purely mathematic nor systematically processed and encompasses immeasurable elements like human experience, interdisciplinary collaborations, pragmatic considerations, heuristic guidelines, and code considerations that account for assumptions, approximations, simplifications, constructions, material uncertainties, etc. The overarching methodology in this paper is to mimic supervised intellectualism to accommodate classic and overlooked or ignored design variables. The algorithms are baked into a series of if-then and conditional logic to arrive at solutions. Because of the inevitable maze of unanticipated design situations and issues, human live engagement has been considered the natural interface component. To our knowledge, the application of this concept for CLT/DLT is fresh in pertinent literature. Selected examples, including input, outputs, and available manual and other software solutions from the literature, are presented next. The results are satisfactory. This procedure has been shown to be beneficial [28, 29, 30].

Fig. 4 shows the blueprint of the input data and the design process cycles for the first app. A typical pseudocode is as follow: Ⓐ – user interface, define panel geometry and layup, define material, define the design load. Ⓑ – algorithm, calculations. Ⓒ – expert interface, check output performance, passes: y/no – loop, check other requirements, finalize design.

Fig. 5 shows its input parameters dialog box of the solver. In the dialog box in Fig. 5, the objective function is defined. It is the mathematical formula with coefficients x to z that the designer engineer is seeking to meet from the design. While the values of the constraints C1, C2, C3, and C4 are on the right side of the box. They represent the conditions or limitations that a solution of the objective function must satisfy. Using randomly selected mathematical problems from literature ~~and~~ in conjunction with Python and Matlab, this app has been deployed, demonstrated, and validated as shown in Table 3. The selected examples in Table 3 span from linear to optimization cases with various constraint conditions. Many other cases have been successfully tested with the same high precision.

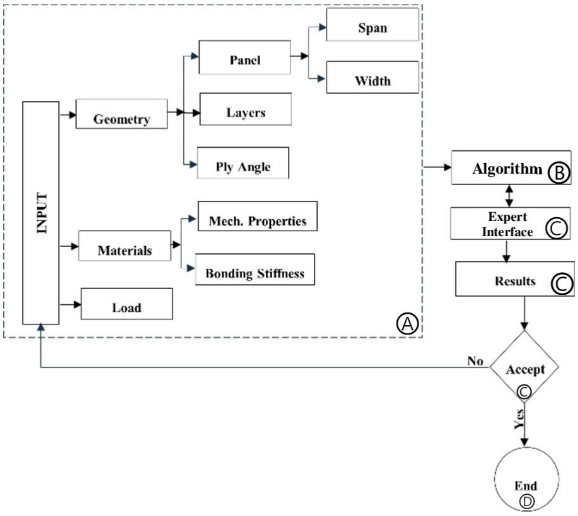


Fig. 4. The blueprint of the input data for applied App for CLT and DLT.

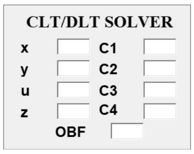


Fig. 5. CLT/DLT Solver Parameters dialog box.

Table 3. Comparison of the CLT/DLT.

Objective Function	Constraints	CLT/DLT Solver	MS Solver ÷ Python, MATLAB, Solver
x1 + x2 + 3 x3 − 0.5 x4 = 0	x1 + 2 x3 <= 740 2 x2 − 7 x4 <= 0 2 x2 − x3 + 2 x4 >= 0.5 x1 + x2 + x3 + x4 = 9	x1 = 0 x2 = 3 x3 = 0 x4 = 6	1
6 x1 + 7 x2 − 280 = 0	2 x1 + 3 x2 <= 120 4 x1 + 2 x2 <= 100 x1 + x2 <= 40 x1 >= 0 x2 >= 0	x1 = 0 x2 = 40	1
max. 2 x1 + 3 x2	x1 >= 0 x2 >= 0 x1 + x2 <= 10 x1 <=5	x1 = 0 x2 = 10	1
max. x1 x2	x1 + x2 <= 10 2 x1 + x2 >= 5 x1 >= 0 x2 >= 0	x1 = x2 = 5	1

7. Summary and Conclusions

Although CLT and DLT applications have proliferated greatly in North America and Europe, the current standards lack essential parameters for structural design. The available engineering specifications provide a wealth of guidelines for their structural design. Nonetheless, the bonding and ply angle are absent. This paper fills this gap from different perspectives.

For having the correct technical understanding, the paper logically reviewed some of the representations in the literature about CLT. The fundamentally unfounded use of Hankinson formula for DLT bending applications was referred to here just as an illustration because it was on the in-plane failure theory and is unrelated to the Bernoulli-Euler theory.

To ascertain the effects of interlayer bonding stiffness and ply angles on the load-bearing performance, the paper presented a rigorous analytical model and solutions that equally work for the CLT and DLT. Because of their versatility, Fourier trigonometric series are used to solve the advanced governing equations. The model and solutions are validated and verified. Pertinent analysis and discussion are conducted to address critical design issues, such as the hypothetical perfectly rigid bonding for laminates, how the choice of constituent species affects the economical design of the panels, and whether there is a boundary between CLT and DLT where one becomes preferred over the other for applications. The results revealed that bonding stiffness must not be underestimated across the board as assumed in the literature~~.~~ Also, in using weak adherent species, a very stiff bonding would not be necessary, and vice versa would be uneconomical. However, the engineer can assume the stiffness at which interlayer deformation becomes practically negligible.

All known pertinent international design specifications include practical formulas for the bending stiffness of the CLT, but a similar formula is absent from the literature for the DLT. This paper developed the missing formula incorporating the bonding, the ply-angle, and hybrid species. It has been tested on a large number of layers that exceeded practical limits.

The absence of a record of studies that covers the scope presented in this paper appealed to take the first step toward a realistic approach with its pseudo-ES methodology. It introduces and demonstrates novelty and innovation through its fresh conceptualization and adaptable application of the design process and architecture, integrating the computational framework and basis for design using the engineer as the natural interface or the interaction bridge for adaptive decision-making capabilities.

To overcome the challenges and cumbersomeness encountered in the iterative design process, the paper automated it with the development of fresh two dedicated stand-alone apps to quickly achieve the final accurate product. They have been validated by solving various types of problems, and the results are compared with manual calculations and literature. They have been coded using high-level programming languages such as Python, Matlab, and VS. It is worth noting that our pseudo-ES is not a calculator. The reader may contact the author for additional details.

References

[1]. N. Mascia, E. Nicolas, R. Todeschini, Comparison between Tsai-Wu failure criterion and Hankinson's formula for tension, *Wood Research*, Vol. 56, Issue 4, 2011, pp. 499-510.

[2]. M. Arnold, Mechanical properties of diagonal laminated timber (DLT) with respect to point-supported mass timber slabs, PhD Thesis, *Technische Universität München*, Munich, 2023.

[3]. Cross laminated timber (CLT), APA – The Engineered Wood Association, 2016, https://www.apawood.org/cross-laminated-timber

[4]. J. Turesson, Diagonal compression of cross-laminated timber, Master's Thesis, *Luleå University of Technology*, Luleå, 2016.

[5]. D. Buck, O. Hagman, Production and in-plane compression mechanics of alternatively angled layered cross-laminated timber, *BioResources*, Vol. 13, Issue 2, 2018, pp. 4029-4045.

[6]. A. Noor, C. Bert, Computational Models for Sandwich Panels and Shells, *American Society of Mechanical Engineers*, New York, 1996.

[7]. E. Karacabeyli, B. Douglas, CLT Handbook, *FPInnovations*, Pointe-Claire, 2013.

[8]. Y. Zhang, L. Zhang, S. Zhang, Exact series solutions of composite beams with rotationally restrained boundary conditions: Static analysis, *Archive of Applied Mechanics*, Vol. 92, Issue 12, 2022, pp. 3999-4015.

[9]. R. Szilard, Theories and Applications of Plate Analysis: Classical, Numerical and Engineering Methods, *John Wiley & Sons*, Hoboken, 2004.

[10]. S. Timoshenko, J. Goodier, Theory of Elasticity, *McGraw-Hill Company*, New York, 1951.

[11]. R. Hussein, Applied AI for DLT and CLT with imperfect bonding, in *Proceedings of the 7th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI)*, 2025, pp. 30-38.

[12]. P. Fazio, R. Hussein, K. Ha, Sandwich beam-columns with interlayer slips, *Journal of Engineering Mechanics*, Vol. 108, Issue 2, 1982, pp. 354-366.

[13]. R. Hussein, Sandwich plates with interlayer slips, *Journal of Engineering Mechanics*, Vol. 110, Issue 4, 1984, pp. 493-506.

[14]. R. Hussein, Orthotropic sandwich plates with interlayer slip and under edgewise loads, *Structural Engineering and Mechanics*, Vol. 17, Issue 2, 2004, pp. 153-166.

[15]. R. Hussein, K. Ha, P. Fazio, Thermal stresses in sandwich panels with interlayer slips, *Journal of Thermal Stresses*, Vol. 12, Issue 2, 1989, pp. 191-207.

[16]. R. Hussein, P. Fazio, K. Ha, Effects of bonding stiffness on thermal stresses in sandwich panels, *Journal of Aerospace Engineering*, Vol. 5, Issue 4, 1992, pp. 480-490.

[17]. R. Hussein, Application of plate theory to laminated wood composites with non-rigid adhesive, *International Wood Products Journal*, Vol. 1, Issue 1, 2010, pp. 35-42.

[18]. C. S. Krishnamoorthy, S. Rajeev, Artificial Intelligence and Expert Systems for Engineers, *CRC Press*, Boca Raton, 1996.

[19]. A. Haftkhani, H. Hematabadi, Effect of layer arrangement on bending strength of cross-laminated timber (CLT) manufactured from poplar, *Buildings*, Vol. 12, Issue 5, 2022, 608.

[20]. S. Timoshenko, D. Young, Theory of Structures, *McGraw-Hill Company*, New York, 1965.

[21]. M. Jeleč, D. Varevac, V. Rajčić, Cross-laminated timber (CLT) – a state of the art report, Journal of the Croatian Association of Civil Engineers, Vol. 70, Issue 2, 2018, pp. 75-95.

[22]. E. Karacabeyli, S. Gagnon, Canadian CLT Handbook, *FPInnovations*, Pointe-Claire, 2019.

[23]. EN 1995-1-1: Eurocode 5 - Design of timber structures, *European Committee for Standardization*, 2004.

[24]. Technical recommendations for construction, *Italian National Research Council*, 2006.

[25]. A. Gustafsson, The CLT Handbook – Facts and Planning, *RISE Research Institutes of Sweden*, Stockholm, 2019.

[26]. S. Timoshenko, S. Woinowsky-Krieger, Theory of Plates and Shells, *McGraw-Hill Book Com*pany, New York, 1959.

[27]. E. Nilsson, Characterization of cross laminated timber properties, Master's Thesis, *Lund University*, Lund, 2021.

[28]. R. Hussein, Knowledge-based tools for monitoring and management, and design of the engineered infrastructure construction systems, in Advances in Computers and Software Engineering: Reviews, Vol. 2 (S. Yurish, Ed.), *IFSA Publishing*, Barcelona, 2019, pp. 199-250.

[29]. R. Hussein, Treatise on sustainable infrastructure construction: Green composites, cross laminated/mass timber, wood truss connectors, nondestructive technologies, health assessment and monitoring: Utility poles and geofoam, in Advances and Technologies in Building Construction and Structural Analysis (I. M. Ammar, Ed.), *IntechOpen*, London, 2021, pp. 1-44.

[30]. R. Hussein, Computer toolkit for the structural analysis and design of cross laminated loadbearing components, Copyright TXu 2-245-722, *United States Copyright Office*, 2020.

(024)

Application of Pseudo-Expert System to Lightweight Synthetic Foam as Load-Bearing Components under Monotonic and Cyclic Loads

R. Hussein
State University of New York (emeritus), Syracuse, NY 13210, USA
Tel.: + 001 6802870342
E-mail: ezpscg@gmail.com

Summary: This paper introduced novelty and innovation via the conceptualization of the hybrid adaptable pseudo-expert system (PES). It demonstrated it in the applied domain of synthetic lightweight foams as load-bearing components for earth-retaining systems. Its app encompassed two attributes: executing code provisions or optimizing the output by enabling the designer to set an objective function that includes some of the immeasurable design parameters. For its application, this study discovered characteristics of the foam that were overlooked or ignored in literature. A small-scale experimental investigation was conducted to identify and understand the missed attributes and their impacts, under three uniaxial monotonic and cyclic compression loads. Incremental loads applied to 35 lightweight polystyrene samples of four-inch cubic sizes arranged in one, two, and three layers. Strain energy, residual strain, stress-strain relations, and sample size were investigated. The discovery showed, unlike what literature reports, that the combined sample size and applied stress affect the performance.

Keywords: Axial compression, Cyclic load, Monotonic load, Pseudo-expert system, Sample size, Synthetic foam.

1. Introduction

Lightweight synthetic materials have a promising future in sustainability. Significant studies have been conducted to enhance the state of the art of foams, and the industry is right to be proud of the essential service it provides but is also seriously overlooking, if not ignoring, opportunities to provide better services in a way that will enhance and not diminish the future. This study brings to the forefront novelty and innovation to some of the overlooked characteristics. Huanhuan Dong [1] provided adequately broad scientific review of foams and applications.

Owing to their favorable characteristics, and real world diverse applications, we aimed at lightweight synthetic rigid foams for construction components. Our studies on load-bearing components and materials started in the seventies. Some of the outcomes were acknowledged by NASA among the top 1.5 % of pertinent accomplishments [2].

This study extends the scope of our previous pertinent ones and the state of the art. It adds novelty and innovation to the existing body of ~~knowledge~~ applied engineering and design of lightweight synthetic foam. It also introduces a genuine hybrid conceptualization of PES app and demonstrates its application. The term "pseudo-static" indicates that the loads are not based on true time-domain history and are derived from incremental static loads, allowing for a simulation of repeated incremental behavior. This small-scale study benefits from this type of loads in a number of ways, including time and cost savings, ease of load application, and simplification of the true cyclic load into a static one.

In our approach to ES, the engineer acts as the natural interface who operates and governs the execution of the system toward reasoning and decision-making. Since our design has been open-ended and the engineer must deal with unforeseeable scenarios, this conceptualization is realistic and imperative.

2. Experimental Investigation

Three incremental static loads with peak values of 40, 80, and 120 pounds were applied to 35 cubic specimens of nominal side length 4 inches each. Commercial lightweight synthetic foam from an anonymous manufacturer with an average density of 13 kg/m^3 was used to prepare the specimens. The specimens were stacked in single, double, and triple layers. Before testing, the specimens were kept for a week in the laboratory at the average ambient temperature of 72 °F. In a typical test, dimensions of each specimen were measured. Axial compression stresses were then applied, and the corresponding displacement was recorded. The axial strain was calculated using the vertical displacement of the machine's moving head. In the cyclic tests, every cycle was carried out three times.

Fig. 1 shows the combined effects of sample size and applied axial stress on residual strain and strain energy. The absorbed energy, represented by the net area under the stress-strain curve, does not return the specimens to the original cubic shape, leaving locked-in residual strain within the specimens even when the applied stress is removed.

The figure shows that the residual strain and strain energy increase with size and applied stress. By changing the sample size from 4-inch to 8-inch under stress of 7.5 psi, the strain and energy increase by 431 % and 165 %, respectively. By increasing the size to 12 inches and the stress to 10 psi, the strain and energy increase by 737 % and 420 %, respectively, compared to the 4-inch size.

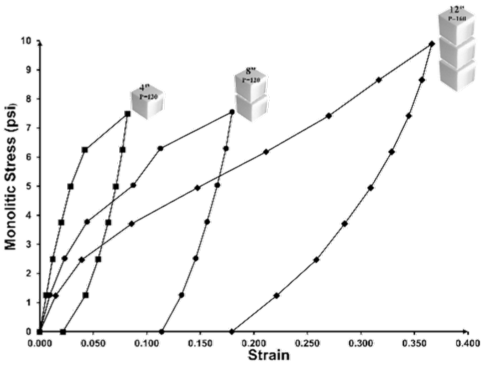


Fig. 1. Stress-stain curves.

Fig. 2 shows the combined effects of sample size and type of applied stress on axial stiffness. Under cyclic stress, the initial modulus for large samples is higher than that of small samples, and the opposite occurs under monotonic stress. This might be due to volume change by closing the voids. For the same size of samples, the cyclic initial stiffness is higher than that of the monotonic one. This discovery means, unlike what literature reports, that the combined size and stress affect deformation in the initial phase. After the initial phase, all specimens continue, if permitted, to deform regardless of size and stress, with less strain in small sizes under cyclic and monotonic stress.

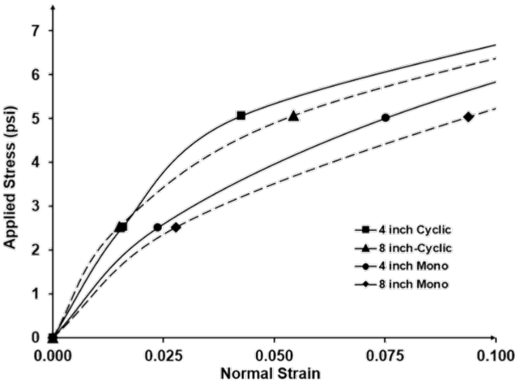


Fig. 2. Effects of size and applied stress.

Fig. 3 shows lateral movement of specimens due to shearing and structural disintegration at interface surfaces between the layers. This should not be misinterpreted as Poisson's ratio effect, which is an intrinsic scientific fundamental property describing the relation between lateral and axial strains in materials.

3. Applications

In this section, PES extends and demonstrates the novelty app for the design of synthetic rigid foam as a load-bearing element in earth-retaining systems. The scientific and engineering fundamentals of earth lateral pressures and forces are basic geotechnical engineering that are beyond the scope of this paper.

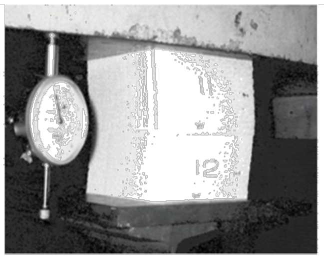


Fig. 3. Shearing of specimens.

On the one hand, each design problem is unique, and only humans can understand not only its ambiguous and unforeseeable scenarios but also think outside its predefined scope(s). On the other hand, construction engineering design is open-ended endeavor, neither processed purely mathematically nor systematically. It encompasses immeasurable parameters, such as experience, interdisciplinary collaborations, pragmatic considerations, heuristic guidelines, and code considerations that account for assumptions, approximations, simplifications, construction issues, and material uncertainties. Thus, a typical design process is iterative. The ethos behind this approach is to improve and refine the concept and plan with each iteration until all standard and specification requirements are met. This approach requires computational tools because it is time-consuming and cumbersome; therefore, digital apps, such as PES, are necessary.

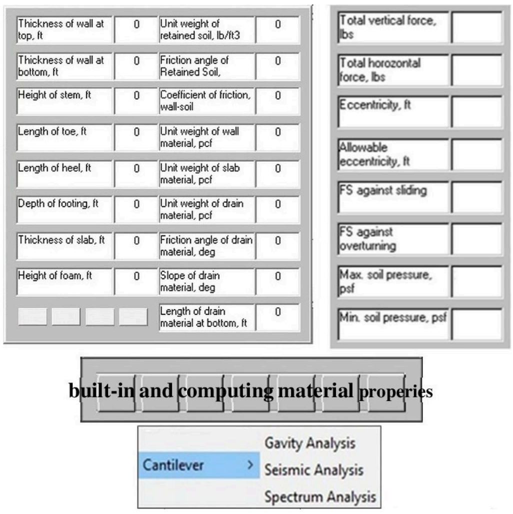
As in other ES systems, PES codifies and applies the explicit standard design-related rules in if-then computable, conditional, and logical statements; constraints, and decision criteria. The algorithm handles the computational framework and determines how the inputs are processed and the outputs are generated. PES encompasses an unmatched component. The engineer can set an objective function using variables such as quantified immeasurable parameters, then optimize it.

The engineer is the inseparable natural interface component who operates and governs the execution of the algorithm and applies human judgment for logical reasoning, interpretation, validation, refinement, and adaptive decision-making outputs. This approach is sensible. Literature has references [3-5] that elaborate on expert systems and provide examples.

Unlike automated AI methods, our concept doesn't codify an automated searchable knowledge base as in AI. And it does not use known designs, reshuffle, reformat, or reorganize existing solutions, and it is not a calculator.

Apps custom-make solutions and designs that fit the data provided. This is our logical and sensible strategy for all our digital development since the seventies. Nonetheless, using trend lines is common in literature. They use regression formulas to fit a curve to the entered data points. Their reliability depends on the quality and quantity of the data. Trendlines assume the data is representative of what it represents. They may over-, under-, or even deceive fittings. As indicated previously, construction engineering encompasses many immeasurable variables. So, what in essence is the dependable usage of trend lines in the type of design at hand? For this reason, we adopted a trustworthy computing method.

Fig. 4 shows our synthetic rigid foam-based dashboard and its parameters dialog box for a typical cantilever-type earth retaining structure. Fig. 5 shows a typical pseudocode of typical design. The dashboard displays the required geometry parameters and mechanical and physical properties of the foam. It provides the option of using built-in physical and mechanical properties, either from available sources or calculated using known formulas. Once the input data is entered, the engineer can edit it, set an optimization objective function, continue the process, or exit. The algorithm will then perform static, seismic, and spectrum analysis. In these categories, the app returns static and dynamic lateral earth pressure coefficients, factors of safety, pressures, and forces. The algorithm has been coded using Python, MATLAB, and MS Visual Studio. All these three software applications can be downloaded for free from the internet. The reader may contact the author for additional details.



Thickness of wall at top, ft	0	Unit weight of retained soil, lb/ft3	0
Thickness of wall at bottom, ft	0	Friction angle of Retained Soil,	0
Height of stem, ft	0	Coefficient of friction, wall-soil	0
Length of toe, ft	0	Unit weight of wall material, pcf	0
Length of heel, ft	0	Unit weight of slab material, pcf	0
Depth of footing, ft	0	Unit weight of drain material, pcf	0
Thickness of slab, ft	0	Friction angle of drain material, deg	0
Height of foam, ft	0	Slope of drain material, deg	0
		Length of drain material at bottom, ft	0

Total vertical force, lbs	
Total horozontal force, lbs	
Eccentricity, ft	
Allowable eccentricity, ft	
FS against sliding	
FS against overturning	
Max. soil pressure, psf	
Min. soil pressure, psf	

Fig. 4. Input, Output, and database dashboards.

4. Summary and Conclusions

Lightweight synthetic materials have a promising future in sustainability. Yet, expert systems' up-to-date development as pertinent design aids is absent from the literature as a load-bearing component in construction. This study closed this gap.

This study discovered characteristics of lightweight polystyrene foam that had been overlooked in previous studies. To determine and understand the overlooked characteristics and their effects, a small-scale experimental study was carried out. Thirty-five four-inch cubic lightweight polystyrene samples were stacked in one, two, and three layers and subjected to three pseudo-uniaxial monotonic and cyclic compression stresses. Strain energy, residual strain, stress-strain relations, and sample size were investigated.

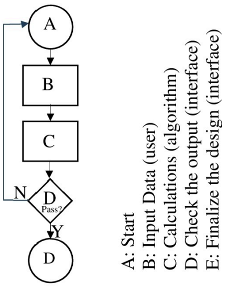


Fig. 5. Flow Chart of typical design.

The results suggested that as sample size and applied stress increase, so do residual strain and strain energy. Large samples have a greater initial modulus than small samples under cyclic stress, whereas the inverse occurs under monotonic stress. The cyclic initial stiffness is greater than the monotonic stiffness for the same size. This discovery indicates that the combined size and stress of samples have an impact on deformation in the initial phase, contrary to what has been reported in the literature. Regardless of size and applied stress, all specimens continue to deform after the initial phase, if allowed, with smaller sizes experiencing less strain under cyclic and monotonic stress. Shearing and structural disintegration at the contact surfaces between the layers cause specimens to displace laterally, and this is not due to Poisson's ratio.

The study introduced novelty and innovation through the conceptualization of the hybrid adaptable pseudo-expert system (PES). The application of synthetic lightweight foams as load-bearing elements for earth-retaining systems was used to illustrate it.

PES uses an if-then computable format to codify and apply design-related rules, conditional and logical statements, predefined constraints, and decision criteria. The computational framework is managed by the algorithm, which also decides how the inputs are handled and the outputs are produced.

PES included two new attributes: either improving the result by allowing the designer to specify an objective function that incorporates some of the immeasurable factors or executing code provisions. It also offers the choice of utilizing built-in mechanical and physical characteristics, which can be computed using established formulas or obtained from accessible sources.

The engineer is the inseparable natural interface component who operates and governs the execution of the algorithm and applies human judgment for logical reasoning, interpretation, validation, refinement, and adaptive decision-making outputs. This approach is sensible. Once the input data is entered, the engineer can edit the data, set the optimization objective function, continue with the execution, or exit.

For the design of gravity, seismic, and spectrum analysis of typical earth-retaining systems that use lightweight, stiff synthetic foam, PES demonstrated the application of the pseudo-hybrid conceptualization. Its app returns earth pressures and forces, factors of safety, and static and dynamic lateral earth pressure coefficients in these categories.

References

[1]. H. Dong, et al., A review of polyurethane foams for multi-functional and high-performance applications, *Polymers*, Vol. 16, Issue 22, 2024, 3182.

[2]. A. Noor, C. Bert, Computational Models for Sandwich Panels and Shells, *American Society of Mechanical Engineers*, New York, 1996.

[3]. R. Hussein, Treatise on sustainable infrastructure construction: Green composites, cross laminated/mass timber, wood truss connectors, nondestructive technologies, health assessment and monitoring: Utility poles and geofoam, in Advances and Technologies in Building Construction and Structural Analysis (I. M. Ammar, Ed.), *IntechOpen*, London, 2021, pp. 1-44.

[4]. R. Hussein, Applications of lignin-based foam as a load-bearing component in engineered laminates, *International Wood Products Journal*, Vol. 15, Issue 2-4, 2024, pp. 128-138.

[5]. C. S. Krishnamoorthy, S. Rajeev, Artificial Intelligence and Expert Systems for Engineers, *CRC Press*, Boca Raton, 1996.

(027)

CT Surface Reconstruction Accuracy under Varying HU Thresholds

Jurica Cvetić and Filip Šuligoj
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lučića 5, 10000 Zagreb, Croatia
Tel.: + 385 99 6750 305
E-mail: jurica.cvetic@fsb.unizg.hr

Summary: Accurate preoperative surface reconstruction is important in image-guided surgery, as registration performance depends on the geometric quality of surfaces reconstructed from computed tomography (CT) scans. This study investigates the effect of Hounsfield unit (HU) threshold selection on head surface reconstruction and its impact on registration accuracy. Outer head skin surface point clouds were generated using a surface reconstruction method across a range of HU thresholds and registered to intraoperative data. Accuracy was evaluated through geometric analysis and target registration error (TRE) measurements on two anthropomorphic head phantoms. HU variation led to systematic geometric changes, with higher thresholds causing inward shrinkage and lower thresholds producing outward expansion. For Phantom 1, TRE generally decreased as the HU threshold was lowered, with the average TRE curve reaching a minimum of 1.18 mm at −700 HU. In contrast, Phantom 2 exhibited a non-linear TRE trend with higher variability across repeated registrations. These results indicate that HU threshold selection influences registration accuracy, but does not provide a reliable or directly predictable indicator of TRE.

Keywords: Hounsfield units, Computed tomography, Surface reconstruction, Image-guided surgery.

---

1. Introduction

In image-to-patient registration for robotic-assisted and image-guided neurosurgery, accurate alignment between preoperative and intraoperative data is important for reliable surgical guidance. Computed tomography (CT) imaging is commonly used to derive three-dimensional head surface models, which are subsequently registered to intraoperative point clouds acquired using depth sensing systems. In these pipelines, the outer head surface is extracted by thresholding CT intensities, thereby defining the geometric representation of the preoperative model. Surface-based registration is typically formulated as the estimation of a rigid transformation $g = (R, t) \in SE(3)$ that maps a source point set $X \subset \mathbb{R}^3$onto a target set $Y \subset \mathbb{R}^3$. The accuracy of this transformation depends on several factors, among which the geometric fidelity of the extracted surface also plays a role. Variations in Hounsfield unit (HU) threshold can systematically alter the reconstructed surface geometry, potentially affecting downstream registration accuracy. This motivates a closer investigation of the relationship between HU threshold selection, surface reconstruction, and registration performance.

Segmentation threshold selection has been widely recognized as an important factor affecting the accuracy of CT- and cone-beam computed tomography (CBCT)-derived surface reconstructions. Dong et al. investigated the influence of HU threshold selection and voxel size on CBCT-based mandibular reconstruction. They showed that reconstructed volumes can deviate from ground truth depending on the selected threshold and acquisition resolution, with different voxel sizes requiring different HU values for minimizing geometric error [1]. Friedli et al. further showed that varying segmentation thresholds alters both the geometry and spatial positioning of cranial base surface models reconstructed from CT and CBCT data, with deviations increasing as the threshold moves away from a reference value [2].

Several studies have additionally shown that the effect of threshold selection is also dependent on acquisition conditions and reconstruction settings. Pauwels et al. reported that HU values and reconstructed intensities vary significantly across CBCT systems and are influenced by acquisition parameters, limiting the use of fixed threshold values [3]. Ghamri et al. confirmed that segmentation settings substantially affect the accuracy of reconstructed craniofacial surfaces from CT data [4]. Zhang et al. demonstrated that both threshold selection and voxel size significantly influence geometric accuracy in CBCT-based tooth reconstruction [5]. Furthermore, Razi et al. showed that gray values in CBCT are not directly equivalent to CT HU values and vary across imaging systems, indicating that HU-based thresholds cannot be treated as standardized across different platforms [6].

Beyond geometric reconstruction, several studies have investigated how segmentation quality influences downstream registration procedures. Park et al. demonstrated that segmentation thresholds affect registration accuracy when aligning CBCT-derived models with optical scans, with optimal performance observed at intermediate threshold values [7]. Andruch and Malecki similarly reported that threshold-dependent segmentation quality directly influences surface registration accuracy [8]. Lo Russo et al. further showed that segmentation parameters impact multimodal registration accuracy, with threshold variations introducing measurable deviations in the final alignment [9].

Although previous studies consistently demonstrate that HU threshold selection influences reconstructed surface geometry and can affect registration quality, the relationship between HU-induced geometric deviations and target registration error (TRE) in image-guided intervention workflows remains insufficiently characterized. In particular, it remains unclear whether systematic surface deviations caused by varying HU thresholds translate into predictable changes in registration accuracy, or whether other factors within the acquisition and registration pipeline dominate the final TRE performance.

In this work, we analyze the influence of HU threshold selection on CT reconstructed surfaces and its impact on registration accuracy. While our previous work [12] introduced and validated a novel surface reconstruction algorithm, the present study contributes a systematic analysis of HU threshold sensitivity and its influence on registration accuracy. We systematically vary the HU threshold and analyze the resulting surface changes using geometric metrics. These geometric variations are then related to TREs calculated using ground truth marker positions. The results reveal a consistent geometric transition with HU threshold, yet demonstrate that this does not translate into a predictable change in registration accuracy, suggesting that other pipeline factors are dominant.

2. Materials and Methods

Two anthropomorphic head phantoms [10, 11] with 3D printed attached markers are used in this study. The phantoms were securely fixed using a DORO QR3 Multi-Purpose Skull Clamp attached to the TruSystem 3000H SBS EU operating table, ensuring rigid positioning throughout the experiment. CT data of the phantoms were acquired using a Siemens Somatom Emotion scanner (Siemens Healthineers, Erlangen, Germany) with a slice thickness of 0.625 mm serving as the preoperative dataset. Intraoperative surface data were acquired using an RGB-D camera (Ensenso B57) mounted on a robotic arm (KUKA LBR iiwa 14 R820). Target marker trajectories were planned using 3D Slicer (v5.10.0). All processing and analysis were implemented in C++ and performed on an HP Z4 G4 workstation running Ubuntu 22.04.5 LTS. The experimental setup is presented in Fig. 1.

CT volumes were imported into 3D Slicer, and soft tissue segmentation was performed using the HU threshold preset available in the official 3D Slicer GitHub repository, https://github.com/Slicer/Slicer (accessed on 8th April 2026). Using the *Markups* module, trajectories were planned for a total of nine markers, seven located on the superior cranial region and two in the neck region on both phantoms. The planned trajectories were subsequently transferred to the intraoperative space through registration of the preoperative and intraoperative data.

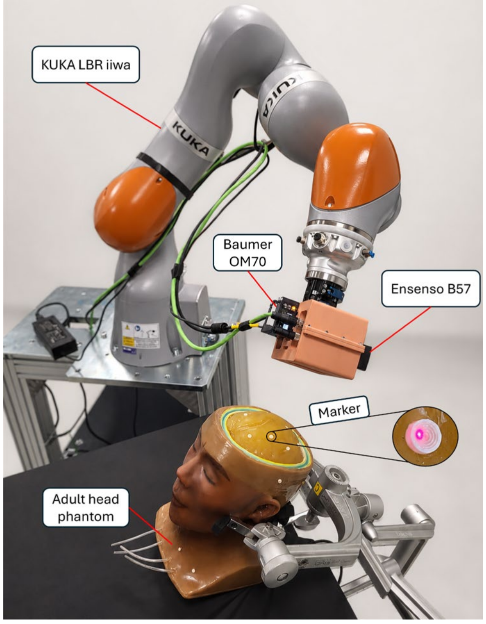
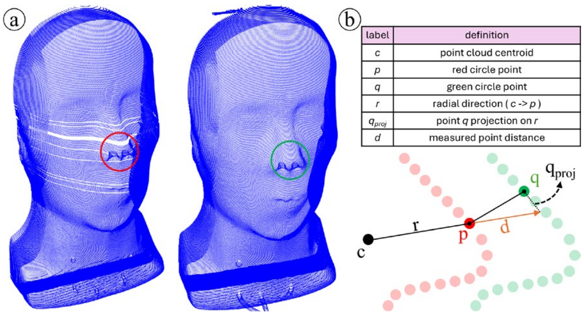


Fig. 1. Experimental setup. A 7-axis collaborative robot, the KUKA LBR iiwa, is used to manipulate a tool that consists of: (1) an Ensenso B57 RGB-D camera used to acquire intraoperative data in the form of a phantom face point cloud for registration with the preoperative data; (2) a Baumer OM70 laser sensor used to measure the TRE. Two anthropomorphic adult head phantoms are used in the experiments, one of which features an openable and removable calvaria.

CT surface point clouds were generated using a surface reconstruction method based on our previous work [12]. The CT volumes were thresholded using a lower HU varying value T and an upper threshold fixed at +220 HU to obtain outer skin contours from axial, sagittal and coronal directions to form a unified surface point cloud. To analyze the influence of HU threshold selection, the reconstruction was repeated for multiple values with steps of 50 in the range −200 HU to −700 HU, resulting in 11 distinct point clouds, two of which are visualized in Fig. 2. This range was chosen based on empirical results in combination with [13]. Only the lower HU threshold was varied, while the upper bound was kept fixed, since higher HU values correspond to internal structures and do not significantly influence outer surface extraction. The generated point clouds were used for registration with intraoperative data obtained using an RGB-D camera. Point clouds were registered in two stages. Coarse registration was performed via singular value decomposition (SVD) using manually selected corresponding point pairs, with all selections made by the same operator across consistent anatomical regions - the two eye pupils, nasal root, and midpoint of the lips. While exact repeatability of manual point selection cannot be guaranteed, the coarse alignment consistently provided a sufficiently close initial transformation for the subsequent fine registration stage. Fine registration was then performed using the Iterative Closest Point (ICP) algorithm to refine the alignment, configured with a maximum correspondence distance of 2.0 mm, a maximum of 100 iterations, a transformation epsilon of 1×10^{-10}, and a fitness epsilon of 1×10^{-7}.



label	definition
c	point cloud centroid
p	red circle point
q	green circle point
r	radial direction (c -> p)
q_{proj}	point q projection on r
d	measured point distance

Fig. 2. (a) Two anthropomorphic head phantom surface point clouds generated using lower HU thresholds of −200 and −650, respectively. Visible differences can be observed in surface completeness, with lower HU values introducing additional outer surface layers. (b) Illustration of signed surface displacement computation. For each reference point $\boldsymbol{p}$, the nearest point $\boldsymbol{q}$ in the test cloud is identified. The displacement vector is projected onto the radial direction $\boldsymbol{r}$ defined from the centroid $\boldsymbol{c}$, yielding the signed displacement $\boldsymbol{d}$, which captures inward and outward surface changes.

3. Experimental Results

Evaluation was conducted in two parts: geometric analysis of CT surface point clouds generated under varying HU thresholds, and assessment of registration accuracy using TRE computed at predefined marker locations.

In the geometric analysis, each point cloud was compared to the reference surface (−300 HU). This value was selected based on empirical evaluation, as it offered a visually complete surface representation with minimal reconstruction artifacts and closely matched soft tissue segmentation preset values used in 3D Slicer. Signed surface displacement was calculated by finding, for each reference point, the closest point in the test cloud and measuring the shift along the outward radial direction from the surface centroid, with the final value obtained as the average over all reference points. The results, as shown in Fig. 3a and Fig. 3b, showed a consistent trend across all evaluated thresholds. Higher HU values resulted in incomplete surface reconstruction and inward displacement relative to the reference, while lower HU values produced outward expansion of the surface. This behavior was reflected in the signed distance values, which transitioned from negative at higher thresholds to positive at lower thresholds.

For registration accuracy evaluation, the RGB-D camera’s tool center point was positioned at the tip of each phantom’s nose at a fixed distance of 350 mm, which provided sufficiently detailed intraoperative point clouds. These point clouds were registered to the preoperative point clouds generated using different HU segmentation thresholds. Each registration was repeated three times per HU value. The resulting transformation matrices were used to position the Baumer OM70 laser tool center point along predefined marker trajectories. Ground truth positions were obtained by manually aligning the laser point with the center of each marker. The TRE was then computed for each marker by comparing the ground truth positions with the corresponding positions obtained from the three repeated registrations. The results show that the relationship between HU threshold and registration accuracy is non-monotonic. In Fig. 3a (Phantom 1), all three registrations follow a consistent trend, where the TRE generally decreases as the HU threshold is lowered, with the average curve reaching a minimum of 1.18 mm at −700 HU. In contrast, Fig. 3b (Phantom 2) exhibits a less consistent pattern. The TRE values show greater variability across repeated registrations, and the average TRE curve reaches its minimum of 0.845 mm at -300 HU, without a clear monotonic trend. Individual registrations deviate more significantly compared to Phantom 1. The results demonstrate that HU threshold selection influences the geometry of CT-reconstructed surfaces, yet this geometric variation does not translate into a predictable or monotonic change in registration accuracy. The absence of a direct HU-TRE correlation can be attributed to several factors: uniform surface expansion or shrinkage caused by HU variation preserves the local shape features that surface-based registration primarily relies upon. Furthermore, ICP's sensitivity to initialization introduces run-to-run variability that is independent of the preoperative surface geometry. Also, depth sensor noise in the intraoperative acquisition constitutes an error source that dilutes the effect of HU selection on the final TRE.

4. Conclusions

This work investigated the influence of HU threshold selection on CT-based surface reconstruction and its downstream impact on image-to-patient registration accuracy, evaluated through geometric analysis and TRE measurements on two anthropomorphic head phantoms.

The results confirm that HU threshold induces a systematic and consistent geometric shift in the reconstructed surface, transitioning from inward shrinkage at higher thresholds to outward expansion at lower values. Despite this regularity, the relationship between HU threshold and TRE was found to be non-monotonic and phantom-dependent, with optimal registration accuracy achieved at different thresholds for each phantom. The results demonstrate that HU threshold is a contributing but non-dominant factor within the registration pipeline, whose effect on TRE is mediated and often obscured by ICP initialization sensitivity, intraoperative sensor noise, and dataset-specific surface characteristics. Consequently, selecting a fixed HU value for surface extraction is insufficient in practice, and per-case adaptive threshold selection should be considered.

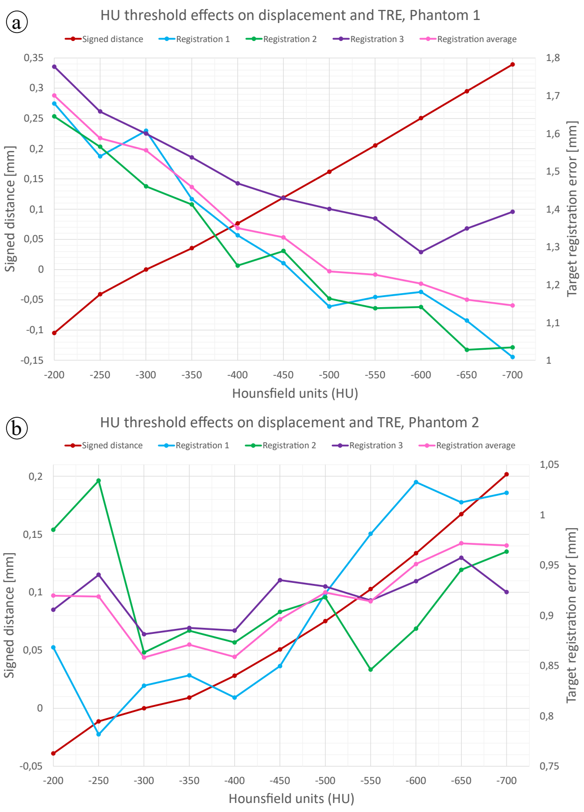


Fig. 3. Mean signed surface displacement (left vertical axis) and target registration error (TRE, right vertical axis) as a function of the HU threshold. (a) Phantom 1: the red curve represents the mean signed surface displacement, where negative values indicate inward surface shrinkage and positive values indicate outward expansion. Lowering the HU threshold value produces a transition from shrinkage to expansion. The blue, green, and purple curves show TRE values obtained from three independent registration sequences (33 registration matrices in total), all exhibiting a decreasing TRE trend with lower HU threshold values. The pink curve represents the mean TRE across all three sequences. (b) Phantom 2: the signed surface displacement follows the same overall trend as in Phantom 1. However, the TRE curves exhibit a less consistent, non-linear behavior across HU thresholds. The averaged TRE curve (pink) indicates a different registration response compared to Phantom 1.

Future work will extend the analysis to a wider range of datasets and CT systems to better isolate acquisition-dependent effects, and will aim to identify geometric or statistical indicators that more reliably predict registration performance across varying conditions.

Acknowledgment

This research was funded by the project INSPIRATION – non-INvaSive PatIent RegistrATIon for rObotic Neurosurgery (grant no. NPOO.C3.2.R2-I1.06.0153), financed by the European Union through the National Recovery and Resilience Plan (NPOO).

References

[1]. T. Dong, L. Xia, C. Cai, L. Yuan, et al., Accuracy of in vitro mandibular volumetric measurements from CBCT of different voxel sizes with different segmentation threshold settings, *BMC Oral Health*, Vol. 19, Issue 1, 2019, 206.

[2]. L. Friedli, D. Kloukos, G. Kanavakis, et al., The effect of threshold level on bone segmentation of cranial base structures from CT and CBCT images, *Scientific Reports*, Vol. 10, Issue 1, 2020, 7361.

[3]. R. Pauwels, R. Jacobs, S. R. Singer, M. Mupparapu, CBCT-based bone quality assessment: Are Hounsfield units applicable ? *Dentomaxillofacial Radiology*, Vol. 44, Issue 1, 2015, 20140238.

[4]. M. Ghamri, K. Dritsas, J. Probst, et al., Accuracy of facial skeletal surfaces segmented from CT and CBCT radiographs, *Scientific Reports*, Vol. 13, Issue 1, 2023, 21002.

[5]. Y. Zhang, Y. Liu, T. Liu, J. Zhang, et al., Evaluation of CBCT reconstructed tooth models at different thresholds and voxels and their accuracy in fusion with IOS data: An in vitro validation study, *BMC Oral Health*, Vol. 24, Issue 1, 2024, 1571.

[6]. T. Razi, P. Emamverdizadeh, N. Nilavar, S. Razi, Comparison of the Hounsfield unit in CT scan with the gray level in cone-beam CT, *Journal of Dental Research, Dental Clinics, Dental Prospects*, Vol. 13, Issue 3, 2019, pp. 177-182.

[7]. S.-W. Park, R. G. Yoon, H. Lee, H.-J. Lee, et al., Impacts of thresholds of gray value for cone-beam computed tomography 3D reconstruction on the accuracy of image matching with optical scan, *International Journal of Environmental Research and Public Health*, Vol. 17, Issue 17, 2020, 6375.

[8]. K. Andruch, M. Malecki, Surface registration accuracy of clinically obtained intraoral optical scans with manually threshold segmented CBCT data, International Journal of Dental Sciences and Research, Vol. 8, Issue 1, 2020, pp. 7-16.

[9]. L. Lo Russo, M. Lorusso, C. Ercoli, R. Sorrentino, et al., Effect of the CBCT data segmentation threshold on registration accuracy with surface scanning, *The Journal of Prosthetic Dentistry*, Vol. 135, Issue 3, 2026, pp. 595.e1-595.e10.

[10]. Static adult head phantom, True Phantom Solutions, https://www.truephantom.com/product/static-adult-head-for-rd/

[11]. Dynamic adult head phantom, True Phantom Solutions, https://www.truephantom.com/product/adult-head-dynamic-advanced-x-ray-ct-us-mri/

[12]. J. Cvetić, B. Šekoranja, M. Švaco, F. Šuligoj, Triplanar point cloud reconstruction of head skin surface from computed tomography images in marker less image-guided surgery, *Bioengineering*, Vol. 12, Issue 5, 2025, 498.

[13]. V. N. Chougule, A. V. Mulay, B. B. Ahuja, Clinical case study: Spine modeling for minimum invasive spine surgeries (MISS) using rapid prototyping, in *Proceedings of the 10th International Conference on Precision, Meso, Micro and Nano Engineering (COPEN)*, 2017.

(029)

Fuzzy-Neural Control of Industrial Robots for Deburring Tasks

L. Baron
Polytechnique Montréal, Box 6079 stat. CV, Montreal, QC, H3C 3A7, Canada
Tel.: +1 514 340-4711 ext. 4744
E-mail: Luc.Baron@polymtl.ca

Summary: This work addressed redundancy resolution in robotic deburring tasks using a geometric decomposition of the end-effector twist, enabling task execution beyond classical null-space methods that are ineffective for functionally redundant systems. The learned function $h$was enhanced through a Takagi–Sugeno–Kang fuzzy neural network. The network was trained on simulated data derived from a composite cost function combining joint-limit avoidance and manipulability maximization. This enabled the model to learn desirable secondary behaviors while preserving task consistency. Compared with classical resolved motion rate control and mid-joint avoidance strategies, the proposed method produced smoother joint trajectories and improved overall motion quality.

Keywords: Fuzzy-neural control, Redundancy resolution, Industrial robot, Deburring task.

---

1. Introduction

Industrial robots are extensively employed in manufacturing processes such as polishing, welding, deburring, and other tasks requiring limited mobility. These tasks are considered *redundant* because they do not require the full six degrees of freedom (DOF) of the robot end-effector (EE). In such situations, only specific components of the end-effector twist need to be controlled, while the remaining components are irrelevant to task execution and can therefore be exploited to improve robot performance.

The literature on redundancy resolution is extensive. However, most conventional approaches rely on rectangular Jacobian matrices $\boldsymbol{J}$, with more columns than rows ($n > 6$), in order to exploit the null space of the Jacobian matrix [e.g., 1]. In the case of deburring tasks performed by a 6-DOF robot, the null space of $\boldsymbol{J}$ is empty, which prevents the direct application of such methods. To overcome this limitation, a geometric approach was proposed in [2, 3], in which the end-effector twist is directly decomposed in operational space.

In this paper, this unified framework is extended by incorporating a fuzzy neural network to resolve task redundancy. The network enables the selection of suitable solutions within the redundant subspace.

2. Robot Control

The joint displacement $\Delta\theta$ is iteratively solved from the EE displacement $\Delta\boldsymbol{t}$ by mapping it to a joint update $\Delta\theta_k$, which is expected to reduce the task error at iteration $k + 1$, according to

$$\theta_{k+1} = \theta_k + \Delta\theta_k \tag{1}$$

2.1. No Redundancy

When $n_o = n = 6$, the Jacobian matrix $\boldsymbol{J}$ is square, and the joint update $\Delta\theta_k$ is obtained as

$$\Delta\theta_k = \boldsymbol{J}^{-1}\Delta\boldsymbol{t}, \tag{2}$$

provided that $\boldsymbol{J}$ is full rank.

2.2. Manipulator Redundancy

When $n > 6$, the Jacobian matrix $\boldsymbol{J}$ is rectangular, and the joint update is computed by the traditional null-space projection technique as

$$\begin{gathered} \Delta\theta_k = (\boldsymbol{J}^{+})\Delta\boldsymbol{t} + (\mathbf{I} - \boldsymbol{J}^{+}\boldsymbol{J})\,\mathbf{h}, \\ \boldsymbol{J}^{+} \equiv \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{J}^{T})^{-1}, \end{gathered} \tag{3}$$

where $\mathbf{h}$ is an arbitrary vector of the joint space. The first term of (3) provides the minimum-norm solution of (2) and ensures the accomplishment of the primary task. The second term generates self-motion within the null space of $\boldsymbol{J}$, allowing secondary objectives to be achieved without affecting the primary task.

2.3. Task Redundancy

For a given task, the EE twist can be decomposed into the task-relevant component and a component lying in the redundant subspace, such as

$$\boldsymbol{t} = \boldsymbol{t}^{Task} + \boldsymbol{t}^{Task^{\perp}} = \boldsymbol{T}\boldsymbol{t} + (\boldsymbol{I} - \boldsymbol{T})\boldsymbol{t} \tag{4}$$

Substituting (4) into (2) yields the twist decomposition framework

$$\Delta\theta_k = (\boldsymbol{J}^{+}\,\boldsymbol{T})\Delta\boldsymbol{t} + \boldsymbol{J}^{+}(\boldsymbol{I} - \boldsymbol{T})\,\boldsymbol{J}\,\mathbf{h} \tag{5}$$

The first term generates the joint displacements required to accomplish the primary task, while the second term produces motion within the redundant subspace that does not affect task execution. Unlike the classical formulation in (3), this approach does not require explicit projection onto the null space of $\boldsymbol{J}$; instead, it directly decomposes the EE twist according to task-relevant and task-irrelevant components.

To address joint-limits avoidance, the joint configuration θ is regulated to remain close to the mid-range position $\bar{\theta}$ by minimizing the cost function

$$z = \frac{1}{2}(\bar{\theta} - \theta)^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{W}(\bar{\theta} - \theta) \rightarrow \min, \tag{6}$$

with $\bar{\theta}$ and $\mathbf{W}$ being defined as

$$\bar{\theta} \equiv \frac{1}{2}(\theta_{\max} + \theta_{\min}), \quad \underline{\theta} \equiv \frac{1}{(\theta_{\max} - \theta_{\min})}, \tag{7}$$

where $\mathbf{W} \equiv \mathrm{diag}(\underline{\theta})$ and the gradient $\boldsymbol{\nabla}z$ defines a direction in the joint-space along which the cost function increases. Therefore, to minimize z, the auxiliary vector $\mathbf{h}$ is selected as

$$\mathbf{h} = -\boldsymbol{\nabla}\mathrm{z} = \mathbf{W}(\bar{\theta} - \theta) \tag{8}$$

Computing such a simple vector field is explicite and can be implemented online. For more complex fields, it is faster to precomputer $\mathbf{h}$ from θ and other performance characteristics.

2.4. Fuzzy-Neural Network

The precomputed input-output dataset of the vector field $\mathbf{h}$ can be used to train a Takagi-Sugeno-Kang (TSK) fuzzy neural network [4-5] as

$$\mathbf{h} = \boldsymbol{\Phi}(\theta, \boldsymbol{d}_{limits}), \tag{9}$$

where $\boldsymbol{d}_{limits}$ represents the normalized distance to joint limits. The model partitions the joint space into fuzzy regions and associates each region with a local linear model. The output $\mathbf{h}$ is computed as a weighted combination of these local models, ensuring both smoothness and interpretability. The projection operator $\boldsymbol{J}^{+}(\boldsymbol{I} - \boldsymbol{T})\boldsymbol{J}$ of $\mathbf{h}$ in eq. (5) ensures that the learned contribution remains confined to the redundant subspace and does not interfere with the primary task.

3. Robotic Deburring Tasks

For a deburring task, the EE twist is decomposed using the projection matrix

$$\mathbf{T} = \begin{bmatrix} (\boldsymbol{I} - \boldsymbol{e}\boldsymbol{e}^{T}) & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix}, \boldsymbol{e} = \boldsymbol{R}_{robot}\boldsymbol{R}_{tool}\boldsymbol{k}, \tag{10}$$

where $\boldsymbol{k} \equiv [0,0,1]^{\boldsymbol{T}}$ and $\boldsymbol{R}_{robot}$ denotes the orientation of the robot flange with respect to the robot base, while $\boldsymbol{R}_{tool}$ represents the orientation of the redundant tool axis (illustrated in blue on Fig. 1) relative to the robot flange defined as

$$\mathbf{R}_{\mathrm{tool}} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}, \mathbf{p}_{\mathrm{tool}} = \begin{bmatrix} 0.032 \\ 0 \\ 0.260 \end{bmatrix},$$

with $\phi = 48°$. Table 1 summarizes the Denavit-Hartenberg parameters of the Fanuc M16iB employed to perform the deburring of the four edges of a spherical part of 0.25 × 0.25 *m* of radius 0.3 *m,* and centered at (0, 1, 0.3) *m*.

First, the resolved motion rate (RMR) method of (2) is used as baseline. In this case, the null-space is empty, and therefore, no space is available for joint-limit avoidance. Second, the twist decomposition approach (TWA) defined in (5) is applied, with $\mathbf{h}$ selected according to (8) to enforce joint-limit avoidance. Third, a TSK fuzzy neural network is trained with Gaussian fuzzy sets defined over θ and $\boldsymbol{d}_{limits}$. This representation enhances learning efficiency by explicitly encoding proximity information to the limits.

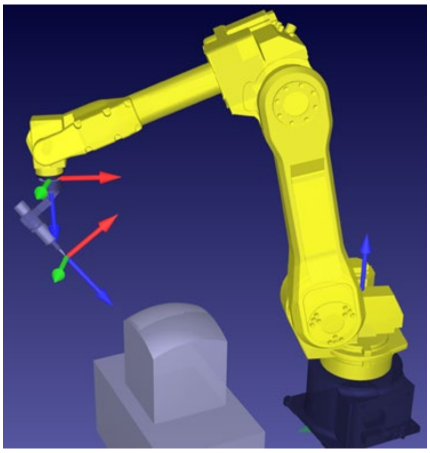


Fig. 1. Fanuc M16iB robot (x = red; y = green; z = blue).

Table 1. DH parameters of the Fanuc M16iB.

joint	θ_i	a_i	b_i	α_i	Min.	Max.
1	θ_1	0.150	0.525	$-\pi/2$	−170	+170
2	θ_2	0.770	0	0.0	−90	+160
3	θ_3	0.100	0	$\pi/2$	−170	+290
4	θ_4	0	0.740	$-\pi/2$	−200	+200
5	θ_5	0	0	$\pi/2$	−140	+140
6	θ_6	0	0.100	0	−450	+450
unit	*rad.*	*m*	*m*	*rad.*	*deg.*	*deg.*

In order to avoid ill-conditioned behaviors of the manipulator, the vector field $\mathbf{h}$ is precomputed as a weighted combination of two cost functions to be minimized such as

$$\mathbf{h} = -\boldsymbol{\nabla}(\alpha\, \mathrm{h}_{\mathrm{limits}} + \beta \log \mathrm{w}) \tag{11}$$

The joint-limit cost function is expressed as

$$h_{limits} = \sum_i \left(\frac{\theta_i - \bar{\theta}_i}{\underline{\theta_i}}\right)^2, \tag{12}$$

while the manipulability log is defined as

$$log\, w = \sum_i log(\sigma_i + \varepsilon), \tag{13}$$

where σ_i is the singular values of $\boldsymbol{J}$ and $\varepsilon = 10^{-12}$. The computation of the manipulability log and its gradient, required to generate high-quality training data, is computationally intensive and may take several hours for the present numerical example.

However, once the parameters of the TSK fuzzy-neural network learned, the network is able to provide a fast approximation of the vector field **h**, making it suitable for real-time implementation. Fig. 2 illustrates the deburring trajectory obtained using TWA in conjunction with the TSK fuzzy neural network.

Figs. 3 and 4 illustrate the joint trajectories during the deburring of the four edges of the part. The RMR control (orange curve), which does not incorporate joint-limit avoidance, fails to complete the task, as the limits of θ_4 and θ_6 are exceeded. In contrast, the TWA control (blue curve), combined with the mid-range joint avoidance strategy, enables successful task execution by inducing rotation of the tool about its redundant axis. However, this rotation is relatively abrupt and lacks smoothness, as it does not explicitly account for proximity to kinematic singularities or unfavorable robot configurations.

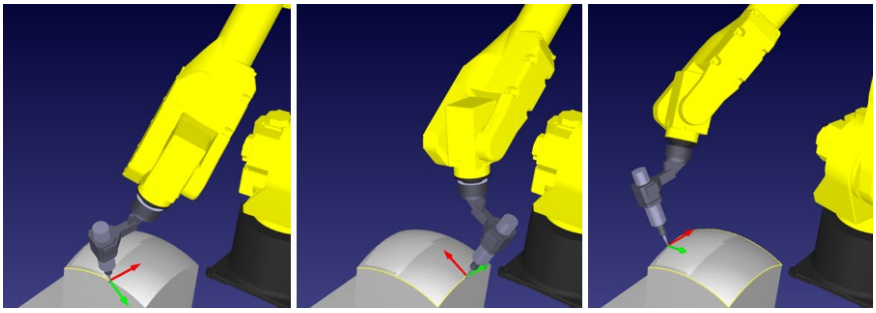


Fig. 2. Edge deburring trajectory with $\alpha = \beta = 1$.

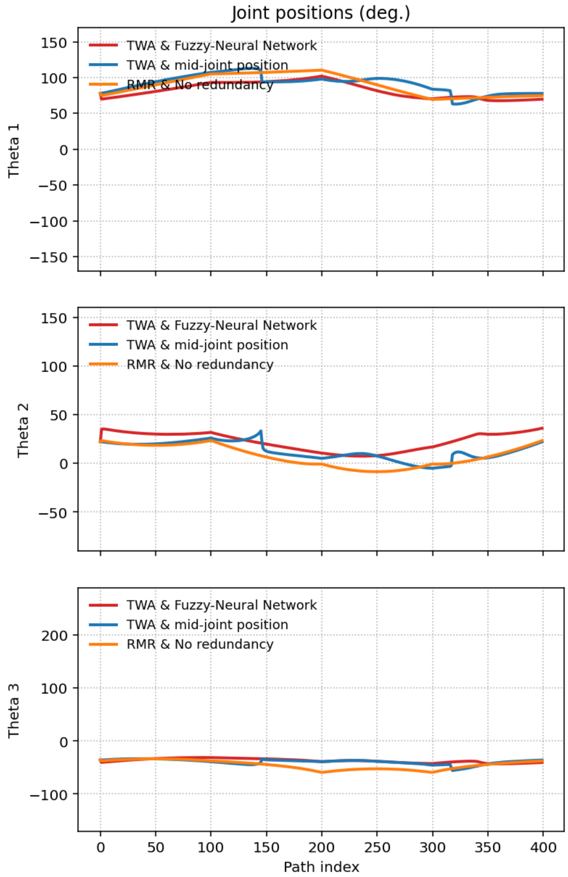


Fig. 3. Joint 1, 2 and 3 positions along the edge.

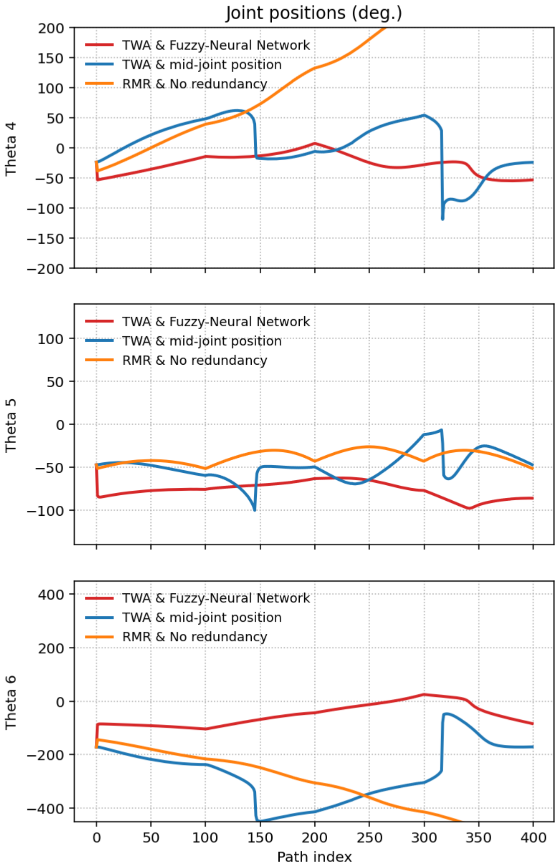


Fig. 4. Joint 4, 5 and 6 positions along the edge.

The TWA control (red curve), augmented with the TSK fuzzy neural network, also induces rotation of the tool about its redundant axis to successfully complete the deburring task. In this case, the resulting trajectory is significantly smoother than that obtained with the mid-range strategy. This improvement stems from the inclusion of manipulability considerations in the training dataset used to compute $\mathbf{h}$.

As indicated in (5), the quantity of interest is not $\mathbf{h}$ itself, but rather the product $\boldsymbol{J}\mathbf{h}$. Learning $\boldsymbol{J}\mathbf{h}$ would present addtional advantages, as its dimension is limited to 6—corresponding to the operational space – rather than the number of robot degrees of freedom. This improvement would therefore be more compact and directly related to the task space, making it physically meaningful.

4. Conclusions

This paper presented a unified framework for resolving task redundancy in robotic deburring operations without relying on classical null-space formulations. The proposed approach is based on a geometric decomposition of the end-effector twist into task-relevant and redundant components, enabling redundancy handling even in cases where the Jacobian null space is not explicitly available.

To improve secondary motion generation, a task-weighted formulation was introduced and further enhanced through a learning-based strategy. In particular, a TSK fuzzy neural network was trained to approximate the redundant motion component h, using simulated data derived from a composite objective function that incorporates both joint-limit avoidance and manipulability maximization. This formulation allows the learning process to encode meaningful geometric and kinematic priors while reducing computational cost at runtime.

The results obtained on a 6-DOF industrial manipulator demonstrated that the proposed learned controller improves motion smoothness and maintains task feasibility compared to both the classical resolved motion rate control and the mid-range joint avoidance strategy. Unlike these approaches, the learned model implicitly captures desirable motion behaviors, including joint-limit avoidance and improved manipulability, while ensuring real-time performance.

Future work will focus on extending the framework to more complex contact-rich tasks and incorporating explicit constraints for obstacle avoidance and dynamic effects.

Acknowledgements

We acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada under grant RGPIN-203618.

References

[1]. F. Flacco, A. De Luca, O. Khatib, Control of redundant robots under hard joint constraints: Saturation in the null space, *IEEE Transactions on Robotics*, Vol. 31, Issue 3, 2015, pp. 637-654.

[2]. L. Huo, L. Baron, The self-adaptation of weights for joint-limits and singularity avoidances of functionally redundant robotic task, *Robotics and Computer-Integrated Manufacturing*, Vol. 27, Issue 2, 2011, pp. 367-376.

[3]. L. Baron, D. Chablat, The twist decomposition of serial robots under lower-mobility tasks, in *Proceedings of the International Symposium on Advances in Robot Kinematics (ARK)*, 2026 (in print).

[4]. N. Hassantabar, Automatic kinematic resolution of redundant planar manipulators using Adaptive Neuro-Fuzzy Inference System (ANFIS) modeling, Master's Thesis, *Polytechnique Montréal*, Montreal, 2021.

[5]. L. Baron, N. Hassantabar, Better exploiting the redundancy of serial manipulators with fuzzy logic: A planar 3-DOF example, in *Proceedings of the 4th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI)*, 2022, pp. 76-79.

(040)

Applications of Pseudo-Expert System to Lignin- Based Laminates as Load-Bearing Components

R. Hussein
State University of New York (emeritus), Syracuse, NY 13210, USA
Tel.: + 001 6802870342
E-mail: ezpscg@gmail.com

Summary: Although lignin is the second-most-abundant biopolymer resource, load-bearing panels do not utilize it. As a result, there are no technical resources in the literature for engineers designing lignin-based load-bearing panels, nor are there corresponding digital design tools. These shortcomings hinder the potential use of this sustainable material and product. The main goals of this paper are to develop lignin-based engineered laminates and establish an engineering mechanics-based model that accounts for the panels' essential properties, such as weak bonding stiffness, under transverse mechanical and thermal loads. Independent analytical solutions and experimental measurements were used to validate these developments, with all results showing satisfactory agreement. Numerical results indicate that bonding stiffness must not be underestimated when considering the serviceability criteria outlined in engineering design codes. To balance component interactions, experts can now quantify what constitutes sufficiently stiff bonding. Another goal is the reconceptualization of the rules-based expert system concept. This paper integrated a natural expert rather than a hardcoded routine in the applied pseudo-expert system (ES). It should be noted that this study has no relation to generative AI. A pseudo-expert system is a specific type of standard ES system in applied structural engineering, where humans work alongside applications to meet project requirements. It is worth noting that our research on engineered laminates, from fundamental to applied, began in the mid-seventies. This paper extends the previous work. It reconceptualized the rules-based expert system concept and integrated a natural expert rather than a hardcoded routine in the pseudo system. The engineer becomes in full control of the entire system operation. This approach is about a human-centered system.

Keywords: Bonding stiffness, Rigid foam, Interlaminar, Lignin, Load bearing, Biomass-based resource, Sustainability.

---

1. Introduction

An estimated 45 billion tons of biomass, in general, are produced worldwide each year, mostly composed of lignin [1]. Despite being the second most abundant biopolymer resource, lignin is not being used adequately in infrastructure for load-bearing structural components. Eighty percent of future load-bearing engineered products can use this readily available renewable resource [2, 3]. These products are referred to in the literature as stressed-skin panels, sandwich panels, engineered laminates, and structural insulated panels, where different constituent materials are combined to maximize each layer's properties for the overall structural benefit. Additionally, their basic concepts, production methods, and manufacturing processes are well documented [4-9]. Although the development of new materials, products, and technologies is advancing rapidly, it is taking longer for lignin-based foams to see widespread industrial use [10-16]. This paper leads the way in transforming biomass resources into innovative applications. It demonstrates how architects can utilize materials in construction for nonstructural purposes to create load-bearing components, establishing the technical foundation for lignin-based load-bearing panels.

Notably, this author has documented in highly indexed engineering journals [17-27] the rigorous foundation for the imperfect bonding-based engineered composite, which covers a wide range of load-bearing components and loadings such as beams, beam-columns, panels, and plates under mechanical and thermal loads. Although other credible references exist in the literature, a key distinction is the use of serious-type closed-form solutions and weak bonding. By utilizing lignin-based cores and considering actual bonding under mechanical and thermal loads, this work builds on our earlier research. NASA acknowledged our collection of studies as among the top 1.5 % of achievements [28].

The application of lignin to the production of rigid foams has been the subject of numerous investigations. In those efforts [29], lignin was either chemically modified or not. In our opinion, the lignin-based rigid foams offer an opportunity to create the necessary innovative panels. That, however, is dependent upon technical rules and performance evaluation of the products that use that foam. Small-scale tests were conducted in compliance with ASTM D1621-10, ASTM D1623-09, ASTM C273-16, and ASTM C203-05a [3] to characterize the properties of the rigid lignin-based foam. According to that study, samples containing up to 20 % lignin had bonding strengths between 158 and 211 kPa and compressive strengths between 163 and 182 kPa. The density of the rigid foam was closely related to its compressive strength [30]. The resistance to shear failure at the skin-core interfaces increased with the amount of lignin in the core. This finding showed that lignin increases the interlaminar shear resistance where delamination occurs by strengthening the core-skin connection. These findings illustrate that the use of lignin leverages the potential success of the structural panels.

Literature must have realistic analytical models augmented with validation and design tools to help professionals navigate the challenging design process because the scope of lignin-based panels is applied rather than fundamental. The performance of the panels cannot be comprehended or determined for practical applications without such a necessary approach, which would result in the loss of unparalleled prospects for a sustainable future. Therefore, it makes sense to ask how we may stop this unavoidable opportunity loss. Instead of using the common unrealistic perfect bonding-based simulations or relying on assumptions as seen in the literature, this research developed a rigorous, engineering mechanics-based analytical model for lignin-based panels constructed with bonding having real mechanical properties. The formulation and solutions have been analyzed, discussed, and validated. The degree of actual bond rigidity and its effect on the structural performance of lignin-based panels, such as deformations, have been examined.

Since there are no pertinent closed-form solutions, this paper advances and promotes our previous pertinent study [31]. An engineering mechanic-based model and a series-type closed-form solution, including the weak bonding stiffness in panels under transverse and in-plane loads and thermal gradient, were developed, solved, and satisfied using Fourier trigonometric series. The inherited mathematical level of the formulation has led to the development of an applied pseudo-expert app for everyday applications. The study ascertained the effect of bonding stiffness on the performance of the panels. The numerical results indicated that bonding stiffness must not be underestimated when engineering design code mandatory serviceability is considered. What constitutes perfectly rigid bonding is also quantified by professionals to balance the design constituents. These findings establish a strong technical foundation to not only accept the introduction of lignin-based panels in real-world applications but also contribute to tackling unprecedented environmental, energy, and economic challenges. The paper ascertained the effect of bonding stiffness on the performance of the panels. The numerical results indicated that bonding stiffness must not be underestimated when engineering design code mandatory serviceability is considered. What constitutes perfectly rigid bonding is also quantified by professionals to balance the design constituents. These findings establish a strong technical foundation to not only accept the introduction of lignin-based panels in real-world applications but also contribute to tackling unprecedented environmental, energy, and economic challenges.

In general, design engineers rely on drawings, mathematics, and codes. This does not mean design engineers are excused from exploring opportunities beyond these resources. This is where our app becomes the boon to help the design engineer with the vehicle to spearhead possibilities. Our app is not a calculator. It draws the line between designs that are entirely human-centered and those that are entirely generated by a machine, i.e., an AI approach.

The app focuses on the human-centered interface between people and computers rather than hard-coded statements stored in the computer. Our system is not searchable for a solution, and, again, is not a calculator. It is a rule-based reasoning engine that, in conjunction with the engineer, processes the algorithm. In other words, it does not replace the engineer with saved codes but rather empowers him/her with full control of the design. As important, it does not relate to generative AI. This is a fresh concept where humans are driving the design.

2. Analytical Modeling and Solutions

Consider a three-layer panel of length and width of 2a and b, with outer layers, known as skins, of thickness tf each and an inner layer, known as the core, of thickness tc. The panel is subjected to a uniform load of intensity qo, an end moment Mo, an axial force F, and thermal gradient.

The following presumptions underline the present development:

1. Linear elastic materials are used;
2. First-order strain-displacement relations apply because of the small deformations;
3. There is no deformation in the direction of layer thickness because of the constant panel thickness;
4. The skins have a constant temperature;
5. Uniform temperature change Tu across the thickness, and a linear change ±Tg at the skins are considered;
6. The shear flux at the bonding layer determines the amount of interlayer slip. The term "slip" is used in our published studies [17-27] and others [32-37] to represent interlayer realistic deformation.

The equilibrium core and skin elements require that:

$$\frac{\partial \sigma}{\partial \mathrm{x}} + \frac{\partial \tau}{\partial \mathrm{z}} = 0, \tag{1}$$

$$\frac{1}{\mathrm{b}}\frac{\mathrm{dN}}{\mathrm{dx}} - \mathrm{q} = 0, \tag{2}$$

in which σ and τ is the normal and shear stresses in the core, f and c is the subscripts denoting skin and core, respectively. N is a force in the skin, q is the interlaminar stress, x and z = coordinate axes with the origin at the center of the panel's left end.

The stresses σ and τ are related to the core deformations by the following elasticity equations:

$$\sigma = \mathrm{E_c}\,\frac{\partial \mathrm{u_c}}{\partial \mathrm{x}}, \tag{3}$$

$$\tau = \mathrm{G}\left(\frac{\partial \mathrm{u_c}}{\partial \mathrm{z}} + \frac{\partial \mathrm{w}}{\partial \mathrm{x}}\right), \tag{4}$$

in which u_c and w are displacements of a point in the core along the x and z axes, respectively, E_c and G is the elastic and shear moduli of the core material, respectively.

Substituting Eqs. (3) and (4) into Eq. (1) result in the equilibrium of the core in terms of deformation. The following boundary conditions must be met

$$u_c = 0 \text{ at } x = a \text{ and } z = 0, \tag{5}$$

$$\varepsilon_c = \frac{\partial u_c}{\partial x} = 0 \text{ at } x = 0 \text{ and } x = 2a \tag{6}$$

The following Fourier series satisfy the governing equation and boundary conditions

$$u_c = \sum_{n=1,3,..}^{\infty} U_n \cos(\alpha_n x) \sinh(\mu\, \alpha_n z), \tag{7}$$

in which $\alpha_n = \frac{n\,\pi}{2\,a}$ and $\mu = \sqrt{\frac{E_c}{G}}$, U_n is an unknown to be determined by additional considerations.

The compatibility condition at the core-skin interface is expressed in terms of strains [32-36], thus:

$$\epsilon_s = \epsilon_f - (\epsilon_c)_{z = t_c/2} + \Delta\alpha\, T, \tag{8}$$

in which ϵ is strains, s, f, and c refer to the bonding interlaminar shearing, the skin, and the core, respectively, $\Delta\alpha = \alpha_f - \alpha_c$ where α is the thermal expansion coefficient, T is the temperature change.

The core and skin strains are calculated as follows

$$\varepsilon_c = \frac{\partial u_c}{\partial x} = \\ = \sum_{n=1,3,..}^{\infty} - \alpha_n U_n \sin(\alpha_n x) \sinh(\mu\, \alpha_n z), \tag{9}$$

$$\varepsilon_f = \frac{N}{t_f\, b\, E_f}, \tag{10}$$

where E_f= the skin modulus of elasticity.

The interlaminar deformation, Δ, is given by [32-37] $\Delta = \frac{q}{k}$ where k = the bonding stiffness in units of stress. The strain is thus

$$\varepsilon_s = \frac{\frac{d^2 N}{dx^2}}{k} \tag{11}$$

Equations (8-10), and (11) yield

$$\frac{\frac{d^2N}{dx^2}}{k} = \frac{N}{t_f\, b\, E_f} + \\ + \sum_{n=1,3,...}^{\infty} \alpha_n U_n \sin(\alpha_n x) \sinh(\mu\, \alpha_n t_c/2) + \\ + \Delta\alpha\, T \tag{12}$$

In solving Eq. (12), the boundary conditions to be satisfied by N are

$$N = \frac{F}{2} \text{ at } x = 0 \text{ and } x = 2a, \tag{13}$$

$$\frac{dN}{dx} = 0 \text{ at } x = a \tag{14}$$

Thus, the complete mathematical solution of Eqs. (13), (12) that satisfies Eqs. (14) and (13) is found as

$$N = \frac{F}{2} \cosh\gamma x - \frac{F}{2} \tanh\gamma a \sinh\gamma x - \\ - \sum_{n=1,3,..}^{\infty} N_n U_n \sin\alpha_n x, \tag{15}$$

in which

$$\gamma = \sqrt{\frac{k}{b\, t_f\, E_f}}, \tag{16}$$

$$N_n U_n = \frac{U_n\, [\alpha_n \sinh\left(\mu\, \alpha_n \frac{t_c}{2}\right)] \pm \Delta\alpha\, \Delta T\, \frac{4}{n\,\pi}}{\frac{\alpha^2}{k} + \frac{1}{t_f\, b\, E_f}} \tag{17}$$

The resisting bending moment can now be calculated using the following engineering mechanics equation

$$M_r = \int_{-tc/2}^{+tc/2} \sigma_c\, z\, b\, dz + N_f\, h, \tag{18}$$

in which $h = t_c + t_f$. Equations (19), (18), in conjunction with Eqs. (3), (7), (6), and (16), (15), yield

$$M_r = -b\, E_c \sum_{n=1,3,..}^{\infty} \Bigg((U_n \left[\frac{t_c \cosh\left(\mu\, \alpha_n \frac{t_c}{2}\right)}{\mu} - \right. \\ \left. - \frac{2 \sinh\left(\mu\, \alpha_n \frac{t_c}{2}\right)}{\mu^2\, \alpha_n} \right] + \left[\frac{F\, \alpha\, h}{a\, (\gamma^2 + \alpha^2)} - \right. \\ \left. - h\; N_n\, U_n \right] \Bigg) \sin\alpha_n x \tag{19}$$

To determine U_n, assume the deflection curve as

$$w = \sum w_n \sin\alpha_n\, x \tag{20}$$

Equations (21), (20) satisfies the boundary condition; w = 0 at x = 0 and x = 2a. The applied bending moment due to the load qo, end moment Mo, and axial load F is calculated using the classic theory of structural analysis as follows

$$M_{applied} = qo \left(a\, x - \frac{x^2}{2} \right) + Mo + F\, w, \tag{21}$$

$$M_{applied} = \sum_{n=1,3,...}^{\infty} [\, qo \frac{2}{a\, \alpha_n^3} + \\ + Mo \frac{2}{a\, \alpha_n} + F\, w_n] \sin\alpha_n\, x \tag{22}$$

The applied and resisting moments must be equal. By equating the coefficients of the $\sin\alpha_n\, x$ in Eqs. (20), (19) and (23), (22) yields

$$U_n = \frac{U_{n1} + U_{n2}}{D_1 - D_2}, \tag{23}$$

$$w_n = \frac{w_{n1} + w_{n2}}{D_1 - D_2}, \tag{24}$$

in which

$$U_{n1} = F\left[\frac{-F\,\alpha_n^2}{ab\,G\,(\gamma^2+\alpha_n^2)} + \frac{\alpha_n\,\Delta\alpha\,T\,\frac{4}{n\,\pi}}{G\,(\frac{\alpha_n^2}{k}+\frac{1}{t_f\,b\,E_f})}\right], \tag{25}$$

$$\begin{split} U_{n2} = \left[\frac{F\,\alpha_n^2\,h}{a\,(\gamma^2+\alpha^2)}\right] + \left[\frac{h\,\alpha_n\,\Delta\alpha\,T\,\frac{4}{n\,\pi}}{\frac{\alpha_n^2}{k}+\frac{1}{t_f\,b\,E_f}}\right] - \\ - \left[\frac{2\,q}{a\,\alpha_n^2} + \frac{2\,Mo}{a}\right], \end{split} \tag{26}$$

$$\begin{split} w_{n1} = \Bigg[\frac{h\,\alpha_n\,\sinh\left(\mu\,\alpha_n\frac{t_c}{2}\right)}{\frac{\alpha_n^2}{k}+\frac{1}{t_f\,b\,E_f}} + \\ +b\,E_c\Bigg[\frac{tc\,\cosh\left(\mu\,\alpha_n\frac{tc}{2}\right)}{\mu} - \\ - \frac{2\,\sinh\left(\mu\,\alpha_n\frac{tc}{2}\right)}{\mu^2\,\alpha_n}\Bigg]\Bigg]\Bigg[\frac{F\,\alpha_n^2}{ab\,G}\frac{1}{(\gamma^2+\alpha_n^2)} + \\ + \frac{\alpha_n\,\Delta\alpha\,T\,\frac{4}{n\,\pi}}{G\left(\frac{\alpha_n^2}{k}+\frac{1}{t_f\,b\,E_f}\right)}\Bigg], \end{split} \tag{27}$$

$$\begin{split} w_{n2} = -\Bigg[\mu\,\alpha_n\,\cosh\left(\mu\,\alpha_n\frac{t_c}{2}\right) + \\ + \frac{\alpha_n^2\,\sinh\left(\mu\,\alpha_n\frac{t_c}{2}\right)}{G\left(\frac{\alpha_n^2}{k}+\frac{1}{t_f\,b\,E_f}\right)}\Bigg]\,\Big[\frac{F\,\alpha\,h}{a\,(\gamma^2+\alpha^2)} + \frac{h\,\Delta\alpha\,T\,\frac{4}{n\,\pi}}{\frac{\alpha_n^2}{k}+\frac{1}{t_f\,b\,E_f}} - \\ - \left[\frac{2\,q}{a\,\alpha_n^3} + \frac{2\,Mo}{a\,\alpha_n}\right], \end{split} \tag{28}$$

$$\begin{split} D_1 = \frac{h\,\alpha_n^2\,\sinh\left(\mu\,\alpha_n\frac{t_c}{2}\right)]}{\frac{\alpha_n^2}{k}+\frac{1}{t_f\,b\,E_f}} + \\ +b\,E_c\left[\frac{tc\,\alpha_n\,\cosh\left(\mu\,\alpha_n\frac{tc}{2}\right)}{\mu} - \frac{2\,\sinh\left(\mu\,\alpha_n\frac{tc}{2}\right)}{\mu^2}\right], \end{split} \tag{29}$$

$$\begin{split} D_2 = F\,[\mu\,\alpha_n\,\cosh\left(\mu\,\alpha_n\frac{t_c}{2}\right) + \\ + \frac{\alpha_n^2\,\sinh\left(\mu\,\alpha_n\frac{t_c}{2}\right)]}{G\left(\frac{\alpha_n^2}{k}+\frac{1}{t_f\,b\,E_f}\right)}] \end{split} \tag{30}$$

3. Experimental Validation

An experimental study has been conducted to validate the analytical modeling and closed-form solution described previously. A guarded hot box, shown in Fig. 1, was designed and built to measure the strains in the samples. The device operates by creating a controlled temperature difference across a test specimen to observe steady-state heat transfer. It consists of three primary chambers: a cold box, a metering (hot) box, and a guard box that prevents lateral heat loss from the metering chamber. It operates by creating a controlled temperature difference across a test specimen placed between a hot and a cold side. The apparatus maintains a constant temperature on the hot side while monitoring the temperature difference across the specimen.

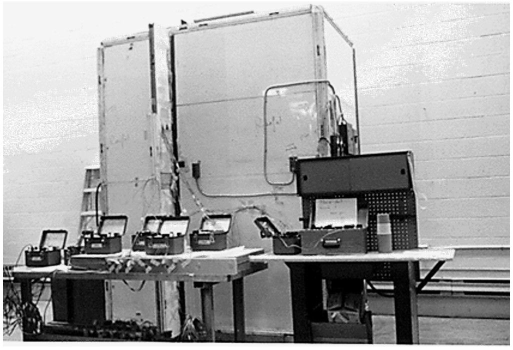


Fig. 1. The guarded hot box.

The hot box is 1.21×1.21×2.43 m, and the metering unit is 0.60×0.91×1.82 m. Two 2000-watt heaters are installed inside the guard unit to vary the temperature relative to the ambient air. The cold chamber is 0.60×1.21×2.43 m and furnished with coils inserted between metal sheets. All the units had thermostats to control the temperatures across the specimens and fans to circulate the temperatures.

The specimens were prepared using aluminum skins bonded to the core using epoxy. The geometrical and material properties were tc = 0.04 m, tf = 0.0635 m, Ef = 68.95 GPa, and α_f = 0.0000234 per °C, Ec = 11.03 GPa, and G = 200 kPa. Strain gauges were used to measure the strains in the skins, and the ambient temperature was measured using thermocouples. The gauges were mounted at 1, 3, 9, 12, and 18 inches from the end of the specimens.

In a typical test procedure, the specimen was inserted in the middle of a partition used between the two chambers. The thermostats were set to the desired temperature, and the strain indicators were initialized. Once the thermostat's setting is reached, the final data from all instruments is read and recorded. The procedure was repeated after 48 hours with new settings.

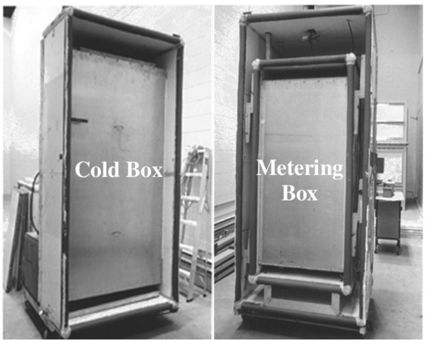


Fig. 2. The cold and hot sides.

The analytical model discussed previously was applied to the geometrical and material properties of the specimens. The analytical and experimental results for the skin's maximal stress are compared in Table 1. It is seen that the analytical findings and the experimental data correspond satisfactorily.

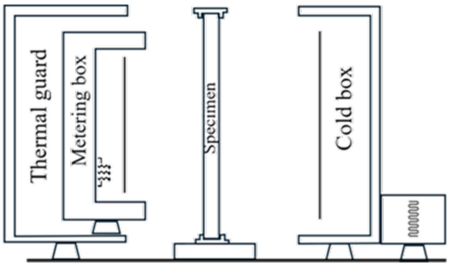


Fig. 3. A sketch of the hot box.

Table 1. Comparison of μ ε, at the center of the samples.

Temperature gradient, ºC	Actual	Closed-Form	Analytic/Actual
-3.92	159.95	161.75	1.01
-4.08	161.05	159.65	0.99
-6.97	123.45	126.05	1.02
-7.28	124.00	125.65	1.01
-9.28	90.00	97.85	1.09

4. Applied Pseudo-ES Approach

From conception to construction, the structural design process is typically cyclical and requires ongoing design refinement. The methodology of this iterative process improves the design with each iteration until all requirements are met. Moreover, engineering design problems are neither solved mathematically nor systematically. Human experience, interdisciplinary teamwork, pragmatic considerations, heuristic advice, and code considerations that account for material uncertainties, assumptions, approximations, simplifications, and constructions are examples of incalculable components of the process. Furthermore, the rigorous analytical modeling that has been described is advanced for regular, everyday actions. Digital computational tools are, therefore, necessary for real-world applications. Our approach is to mimic supervised intellectualism in order to not sacrifice accuracy and efficiency and to save time.

Expert systems are early, foundational artificial intelligence programs designed to mimic human experts' decision-making skills in particular domains. These systems use human-provided rules to solve problems, in contrast to contemporary machine learning or artificial intelligence, which learns from data. Three main components are used by expert systems to function:

(1) Rule Base: in this system, we have procedural and declarative knowledge, e.g. the "how." It includes the algorithm, design codes, etc., principles, methodologies, processes, etc. used to solve problems, IF-THEN statements, allowing the engineer to draw conclusions from known data.

(2) The expert domain where the natural engineer takes the driver seat to drive the entire system. The expert interface becomes the bridge between the user and the rules domain. It is the domain expert where the natural engineer performs the diagnoses, reasoning, evaluation of a scenario(s), decision-making, actions, etc.

(3) User interface: this is the contact point between the user and the system for entering data and receiving results. The user interface is where users interact with the computer. It encompasses the controls for the input data, navigating around, tool tips, etc.

In the context of applied structural engineering, this study extends previous works by re-coining the concept and application of pseudo-ES. This app is about human-computer interaction that focuses on the human-centered interaction. It is a fresh interdisciplinary area that combines human factors and engineering design. The primary goal is to create human-centered, rather than automated hard coded, technology to make decisions. This rigorous structure is shown in Fig. 4. The dashboard's display is divided into sections for the geometry, the applied load, and the material properties. The user can enter the appropriate data, edit it, or reset the display. Once the input data is entered, the algorithm calculates the deflection, for example, using real bonding and perfectly rigid laminates [28]. The natural engineer then takes over the system to ascertain the performance and makes decisions. This approach is not only realistic but also sensible. This algorithm has been coded using Python, MATLAB, and Visual Studio.

5. A Results, Analysis, and Discussion

Analyzing and discussing results obtained from solving realistic panels is crucial to understanding and confidently designing the lignin-based laminates while accounting for the necessary parameters. To this end, consider a panel with a side length of 1.10 m, a skin thickness of 0.032 m, and a core thickness of 0.10 m under a transverse uniform load and a temperature change of -6.67 °C. The lignin core's MOE is 70 × 106 Pa. The skin's MOE is 10.3 GPa. The bonding stiffness varied from 6.89 to 68.95 MPa.

The maximum deflection and skin stress obtained using the series-type closed-form solution presented in this paper are displayed in Fig. 5. The figure illustrates how the deflection is more sensitive to changes in bonding stiffness when it is in the lower range. For example, reducing the bonding stiffness by one order of magnitude from 14 MPa resulted in a 26 % increase in deflection. The equivalent change is less than 1 % because of an order of magnitude drop from 69 MPa/m. It is possible to compute the stress component in the skin under mechanical load with a reasonable degree of inaccuracy by using the popular, perfectly rigid boing-based solutions. In this situation, for example, there is a 4 % variation between the two stresses.

The thermal stress exhibits performance similar to mechanical load deflection. It changed by 27 % due to a decrease in the bonding stiffness by one order of magnitude from 14 MPa.

These findings show that when serviceability is the quantity of interest, bonding stiffness cannot be underestimated. The ratio of bonding stiffness to core stiffness contains the answer. As seen in Fig. 5, the ratio in this example is estimated at 1 or higher. Thus, if the core were too soft, very rigid bonding would not be required, but the contrary would be unwise, according to conventional engineering sense.

The lignin content affects the properties of lignin-based rigid foam, impacting on its physical and mechanical properties. Increasing lignin content improves the Young's modulus, rigidity, and strength but lowers the foam density [38-40]. To improve the benefits of lignin content and avoid poor structure, techniques like chemical modification or solvent fractionation are used. This scientific issue is scientific rather than engineering and falls beyond the scope of this paper.

Fig. 6 illustrates how the use of rigid foam with a high lignin content and bonding stiffness greatly enhanced the span-to-max deflection, for example. The performance of the low-lignin-based foam remained unchanged. This finding suggests that in order to choose the appropriate lignin-based foam for engineering applications, the engineer should confer with the proper material scientists.

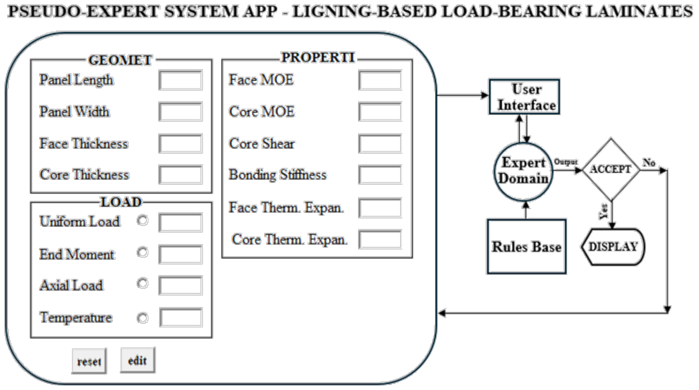


Fig. 4. Flowchart of the pseudo-ES app.

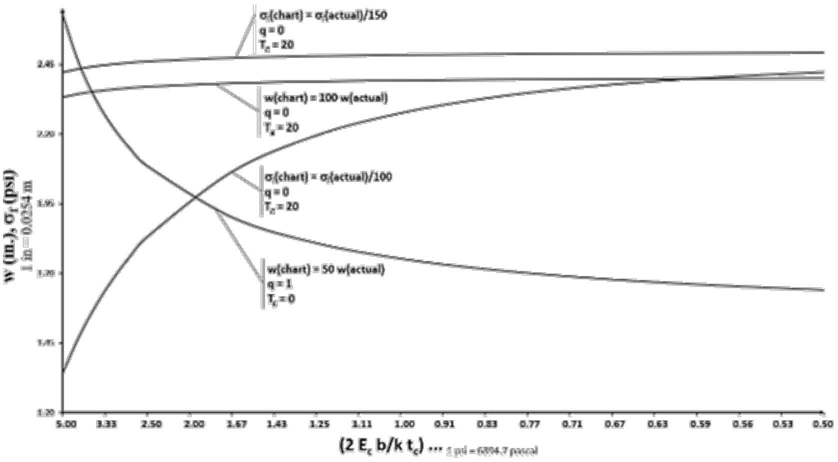


Fig. 5. Effects of bonding.

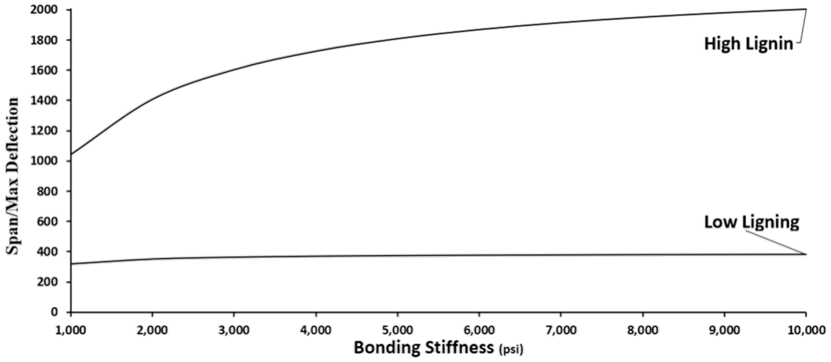


Fig. 6. Effects of lignin content.

6. Summary and Conclusions

Despite being the second most abundant biopolymer resource, lignin is not being used adequately in infrastructure for load-bearing structural components. For sustainable products, this is a missed chance. Because of this, practitioners who would think about using it for load-bearing components lack relevant and sufficient technical resources for engineers to use while developing them. This paper closes this gap by presenting a lignin-based engineered laminates. The paper presented a rigorous analytical model and solutions for various loading scenarios. Because of their versatility, Fourier trigonometric series are used to solve the advanced governing equations. The model and solutions are validated and verified analytically and experimentally. Pertinent analysis and discussion are conducted to address critical design issues, such as the hypothetical perfectly rigid bonding for laminates and how the choice of constituent species affects the economical design of the panels. The findings revealed that while considering serviceability, the bonding stiffness must not be underestimated. The stiffness at which interlayer deformation is essentially insignificant, however, can be assessed by the engineer. The definition of perfectly rigid bonding can also be accurately estimated by engineers. When the bonding becomes so stiff that the change in panel performance can be essentially disregarded, it may be said to be perfectly rigid. These findings give the application of the innovative use of the significantly underutilized resource a strong technological foundation and inspire confidence. Because the structural design process has always been a cyclical one that necessitates constant design improvement and the essential and inseparable live human engagement in the process, a fresh applied pseudo-expert expert system approach to the load-bearing performance of panels is coined to absorb the time-consuming and cumbersome daily routine practice. Its app mimics supervised intellectualism, where humans perform tasks in harmony with computers to navigate the pertinent maze of calculable and uncalculable, required and unforeseen variables to meet the requirements. Its algorithm is coded in high-level language, such as Python, MATLAB, and MS Visual Studio. It is worth noting that this app is neither a calculator nor related to generative AI. The reader may contact the author for additional details.

References

[1]. X. Zhang, et al., Carbon nanostructure of Kraft lignin thermally treated at 500 to 1000 °C, *Materials*, Vol. 10, Issue 8, 2017, pp. 1-14.

[2]. A. Krombholz, N. Eversmann, P. Werner, Lignin based sandwich system for load bearing insulation, in *Proceedings of the World Sustainable Building Conference (World SB14)*, Vol. 2, 2014, pp. 41-47.

[3]. X. Zhang, et al., Lab-scale structural insulated panels with lignin-incorporated rigid polyurethane foams as core, *Industrial Crops and Products*, Vol. 132, Issue 6, 2019, pp. 292-300.

[4]. D. Zenkert, The Handbook of Sandwich Construction, *Engineering Materials Advisory Services*, Warley, 1997.

[5]. A. G. H. Dietz, Composite Engineering Laminates, *The MIT Press*, Cambridge, 1969.

[6]. A. Kausar, et al., State-of-the-art of sandwich composite structures: Manufacturing–to–high performance applications, *Journal of Composites Science*, Vol. 7, Issue 3, 2023, pp. 1-28.

[7]. M. Al-Khazraji, S. Bakhy, M. Jweeg, Composite sandwich structures: Review of manufacturing techniques, *Journal of Engineering, Design and Technology*, Vol. 22, Issue 5, 2024, pp. 1616-1636.

[8]. F. J. Plantema, Sandwich Construction - The Bending and Buckling of Sandwich Beams, Plates and Shells, *John Wiley & Sons*, New York, 1966.

[9]. H. G. Allen, Analysis and Design of Structural Sandwich Panels, *Pergamon Press*, Oxford, 1969.

[10]. J. Kim, et al., Advances and perspectives of using stable isotope probing-based technologies in contaminant biodegradation, *Water Research X*, Vol. 20, Issue 9, 2023, 100187.

[11]. Q. Ji, et al., Comprehensive depolymerization of lignin from lignocellulosic biomass: A review, *Critical Reviews in Environmental Science and Technology*, Vol. 53, Issue 21, 2023, pp. 1866-1887.

[12]. W. Li, J. Shi, Lignin-derived carbon material for electrochemical energy storage applications: Insight into the process-structure-properties-performance correlations, *Frontiers in Bioengineering and Biotechnology*, Vol. 11, 2023, pp. 1-19.

[13]. H. Ye, Customized compatibilizer to improve the mechanical properties of polylactic acid/lignin composites via enhanced intermolecular interactions for 3D printing, *Industrial Crops and Products*, Vol. 205, 2023, 117454.

[14]. P. Bartczak, et al., Green synthesis of chitin/lignin-based polyurethane composites, *Industrial Crops and Products*, Vol. 204, 2023, 117237.

[15]. R. Patel, et al., Transforming lignin into renewable fuels, chemicals, and materials, *Bioresource Technology Reports*, Vol. 22, Issue 6, 2023, 101463.

[16]. A. Lopes, et al., On the path to improve lignin depolymerization and functionalization into bio-based platform chemicals, *Current Opinion in Green and Sustainable Chemistry*, Vol. 43, 2023, 100850.

[17]. R. Hussein, Thermal stress in flat metal-faced sandwich panel, Master's Thesis, *Concordia University*, Montreal, 1978.

[18]. R. Hussein, Structural behaviour of sandwich panels, PhD Thesis, *Concordia University*, Montreal, 1980.

[19]. K. Ha, R. Hussein, P. Fazio, Analytic solution for continuous sandwich plates, *Journal of the Engineering Mechanics Division*, Vol. 108, Issue EM2, 1982, pp. 228-241.

[20]. P. Fazio, R. Hussein, K. Ha, Sandwich beam-columns with interlayer slips, *Journal of Engineering Mechanics*, Vol. 108, Issue 2, 1982, pp. 354-366.

[21]. R. Hussein, Sandwich plates with interlayer slips, *Journal of Engineering Mechanics*, Vol. 110, Issue 4, 1984, pp. 493-506.

[22]. R. Hussein, P. Fazio, Thermal nonlinear behavior of sandwich panels: Experimental measurements, *Experimental Mechanics*, Vol. 25, Issue 6, 1985, pp. 140-144.

[23]. R. Hussein, K. Ha, P. Fazio, Thermal stresses in sandwich panels with interlayer slips, *Journal of Thermal Stresses*, Vol. 12, Issue 2, 1989, pp. 191-207.

[24]. R. Hussein, K. Ha, P. Fazio, Thermal stresses in sandwich plates, *Journal of Thermal Stresses*, Vol. 12, Issue 3, 1989, pp. 333-349.

[25]. R. Hussein, P. Fazio, K. Ha, Analytical evaluations of local failures in sandwich panels, *Building and Environment*, Vol. 26, Issue 2, 1991, pp. 209-215.

[26]. R. Hussein, P. Fazio, K. Ha, Effects of bonding stiffness on thermal stresses in sandwich panels, *Journal of Aerospace Engineering*, Vol. 5, Issue 4, 1992, pp. 480-490.

[27]. R. Hussein, Application of plate theory to laminated wood composites with non-rigid adhesive, *International Wood Products Journal*, Vol. 1, Issue 1, 2010, pp. 35-42.

[28]. A. K. Noor, W. S. Burton, C. W. Bert, Computational models for sandwich panels and shells, *Applied Mechanics Reviews*, Vol. 49, Issue 3, 1996, pp. 155-199.

[29]. S. Gómez-Fernández, et al., Properties of flexible polyurethane foams containing isocyanate functionalized kraft lignin, *Industrial Crops and Products*, Vol. 100, 2017, pp. 51-64.

[30]. A. Paruzel, et al., Rigid polyurethane foam fabrication using medium chain glycerides of coconut oil and plastics from end-of-life vehicles, *ACS Sustainable Chemistry & Engineering*, Vol. 7, Issue 5, 2019, pp. 6237-6246.

[31]. R. Hussein, Applications of lignin-based foam as a load-bearing component in engineered laminates, *International Wood Products Journal*, Vol. 15, Issue 2-4, 2024, pp. 128-138.

[32]. N. M. Newmark, R. J. Siess, I. M. Viest, Tests and analysis of composite beams with incomplete interaction, *Proceedings of the Society for Experimental Stress Analysis*, Vol. 9, Issue 1, 1951, pp. 75-92.

[33]. J. Goodman, E. Popov, Layered beam systems with interlayer slip, *Journal of the Structural Division*, Vol. 94, Issue ST11, 1968, pp. 2535-2547.

[34]. A. Puppo, H. Evensen, Interlaminar shear in laminated composites under generalized plane stress, *Journal of Composite Materials*, Vol. 4, Issue 2, 1970, pp. 204-220.

[35]. J. Goodman, E. Popov, Layered wood systems with interlayer slip, *Journal of the Structural Division*, Vol. 94, Issue 11, 1968, pp. 2535-2548.

[36]. J. Goodman, et al., Composite and two-way action in wood joist floor systems, *Wood Science*, Vol. 7, Issue 1, 1974, pp. 25-33.

[37]. Y. Zhang, et al., Exact series solutions of composite beams with rotationally restrained boundary conditions: Static analysis, *Archive of Applied Mechanics*, Vol. 92, 2022, pp. 3999-4015.

[38]. K. Tihomir, Mechanical, rheological and thermal characterization of lignin-based materials, *LignoCOST Training School*, 2020.

[39]. T. Fagbemigun, C. Mai, Production and characterisation of self-blowing lignin-based foams, *European Journal of Wood and Wood Products*, Vol. 81, 2023, pp. 579-590.

[40]. Q. Yan, et al., Eco-foaming lignin for innovative rigid foam, *Green Chemistry*, Vol. 26, 2024, pp. 5194-5201.

(042)

Classification of Targets from Multi-View Side-Scan Sonar Imagery

O. Katrusha [1], **D. Prylipko** [2] **and K. Yefremov** [1]
[1] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute",
37 Beresteiskyi Ave., 03056, Kyiv, Ukraine
[2] EvoLogics GmbH, Wagner-Régeny-Straße 4, 12489 Berlin, Germany
Tel.: + 380679297090
E-mail: oleksandr.katrusha@gmail.com

Summary: Side-scan sonar is a key sensing modality for underwater applications. In real missions, the same object is typically observed many times from different viewpoints. Most modern automatic target recognition methods process images independently and often struggle to classify them under challenging conditions. This work addresses the target classification problem by aggregating multiple observations of a target. The permutation-invariant Deep Sets architecture is investigated and its performance compared with single-view-based ensemble classifiers. The proposed framework uses a pretrained CNN-based object detector and extends it to a target-level classifier. In addition, a method for generating multi-view, target-oriented side-scan sonar image datasets is proposed. Experiments on a specialized real-world dataset show that multi-view fusion substantially outperforms single-image inference, with a F1 score increase of 2–29 percentage points at the patch level. The Deep Sets approach outperforms ensembles by up to 23 percentage points in target-level F1 score, highlighting its advantage for multi-view target classification.

Keywords: Side-scan sonar, Multi-view classification, Deep learning, Computer vision, Ensemble learning, Deep sets.

1. Introduction

Recent advances in deep learning-based methods have significantly improved the performance of automatic target recognition (ATR) in sonar imagery including target detection and classification [1, 2]. Still, most approaches remain focused on single-image analysis. Our previous experiments with CNN-based side-scan sonar (SSS) object detectors indicate that target classification from individual images remains challenging, especially under adverse conditions and for the images with limited distinctive features.

The observations of the same target may differ substantially due to aspect angle changes, shadow geometry, signal-to-noise variations, and seabed background (Fig. 1). As a result, targets may be clearly recognizable under certain conditions and ambiguous under others, while some seabed clutter may exhibit target-like appearance.

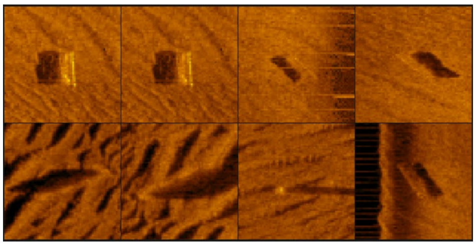


Fig. 1. Eight observations of a single mine from different viewpoints within one mission.

Human operators learn to mitigate such problems by forming a mental representation of a target from its observations. However, the decision-making process based on manual grouping and classification of observations is not practical due to localization errors of various origins and high volumes of data in a typical mission. This motivates research into automated methods that leverage repeated target observations in operational SSS missions and explicitly fuse information from different detections to improve ATR efficiency.

Multi-view classification methods have been in the focus of researchers for years. However, the application of state-of-the-art deep models to multi-view classification of real SSS images still receives little attention. An additional hindrance is the scarcity of publicly available sonar datasets [3] and the absence of target identifiers associated with detections in such datasets.

In this paper, a deep learning-based multi-view target classification method applied to real SSS images is proposed. Three multi-view approaches are considered: two single-view ensembling approaches that use majority vote and the highest average probability for classification, and a target classification model based on Deep Sets method. Deep Sets [4] is a generic set-based deep learning approach that naturally assumes an arbitrary number of observations per object. Any continuous permutation-invariant set function can be represented as element-wise transformation followed by a symmetric aggregation and nonlinear processing. The approach has found numerous successful applications in different domains.

All three approaches are evaluated on real multi-mission SSS data with explicit target-level annotations. The main contributions of this work are:

1. A novel method of multi-view dataset generation from SSS waterfall images and target annotation strategy;

2. Application of permutation-invariant Deep Sets architecture to target-level classification of SSS imagery with a variable number of observations per target;
3. Transfer of a feature extractor trained for object detection to the task of multi-view target classification via ensembling and Deep Sets strategies.

Besides classical ATR missions, the proposed methods can be used in increasingly common swarm operations, 3D reconstruction, SLAM, and advanced target analysis with SSS data.

2. Related Work

A survey work [5] presents multi-view learning as building consistent and shared representations from different sources or viewpoints to use complementary information and improve downstream classification, clustering, and retrieval. Methods are grouped into representation alignment and representation fusion.

It has been experimentally confirmed that the use of multiple representations of the same object or its different physical representations can significantly improve remote ATR performance in remote sensing tasks compared to single-view approaches [6-8].

Teng et al. [7] propose the Sequential Attention Layer-wise Fusion Network (SLFNet) for multi-view classification using graph convolutional networks (GCN). Experiments on multi-view tabular and optical images datasets show that SLFNet outperforms recent classification methods.

A Deep Multi-view Union Learning Network (DMULN) combining canonical correlation analysis (CCA) with a two-branch CNN is proposed in [9] for multi-source image classification. A dedicated view union pooling mechanism aggregates features before classification. Results on standard multi-source remote sensing datasets show improved accuracy over single-view approaches.

Tong and Wang propose Active Multi-View Fusion (AMVF) for synthetic aperture radar (SAR) ATR [8]. It explicitly accounts for weak and redundant correlations between observations by combining sparse representation with spatio-temporal and adversarial losses and reinforcement learning. Results show higher recognition accuracy than traditional multi-view fusion classification methods.

Pei et al. present a multi-view deep learning framework for SAR ATR targeting limited training data by generating synthetic multi-view samples [10]. A convolutional neural network (CNN) with a parallel multi-input architecture learns features per viewpoint and progressively fuses them at different levels. This fusion improves recognition accuracy, validated on the MSTAR dataset.

Williams et al. in their work [11] evaluate multi-aspect classification for ATR on real synthetic aperture sonar (SAS) data by using fundamentally different representations of the same raw sonar data. Experiments on large real marine datasets show improved accuracy over single-view networks even with limited training data and small model capacity.

D'Ales de Corbet et al. [12] investigate multi-view target classification in SAS imagery using CNNs, showing limitations of single-image classification. They compare a single-view CNN, late fusion by averaging predictions, and feature-level fusion by concatenating outputs of parallel branches, where late fusion yields a substantial improvement in classification accuracy.

As for SSS imagery, early work by Zerr et al. [13] demonstrated that a multi-view approach using shadow-derived object profiles and neural-network classifiers can improve classification performance of SSS images by reducing ambiguities of aspect-dependent target appearance, particularly for mine counter-measure (MCM) applications.

Wang et al. [14] also address automatic classification of mine-like objects (MLOs) in SSS images from different viewing directions. Two approaches are proposed: decision fusion using Dempster–Shafer theory and multi-instance logistic regression learning. The use of multiple views improves classification metrics, increases the hit rate for all mine shapes. Evidence fusion based on Dempster–Shafer theory has also been extensively studied in earlier works on multi-view SSS target classification [14-16].

The literature review shows that multi-view and multi-aspect approaches are successfully applied in classification, ATR and remote sensing tasks. They are often implemented based on deep learning models commonly used in modern ATR systems, making it natural to extend them to multi-view settings. At the same time, simple ensemble strategies remain practical due to their ease of implementation and stable performance.

Despite a wide range of existing multi-view methods, there remains a gap in applying modern deep learning approaches to real-world SSS data for multi-view classification (see overview papers [1, 2, 17]). We consider research in this field to be particularly important, as SSS systems are employed much more widely than SAS systems, owing to their lower cost and ease of use.

Among such methods, permutation-invariant architectures like Deep Sets can be built upon existing models, can handle an arbitrary number of unordered observations and aggregate them into a single target representation, which motivates their use in this work to address the gap.

3. Methods and Data

3.1. Methods

In this research, each underwater target comprises an unordered set of manually labelled image patches (views) derived from bounding boxes annotated on SSS waterfall diagrams. In addition to the collection of detection patches, each target is associated with a class label and coordinates. The number of views per target may vary from one to several dozen. Both patches and targets were annotated either as “mine” or NOMBO (non-mine bottom object). The proposed method can be easily extended to a larger number of classes. The classification was applied to the individual patches (patch-level) as well as to targets (target-level). Three multi-view classification approaches are considered:

1. Ensemble of single-view base classifiers using majority vote;
2. Ensemble of single-view base classifiers using the highest average class probability;
3. Multi-view embedding-level aggregation using Deep Sets architecture.

The first approach defines the target class as (1):

$$f_{mv}(X) = \arg\max_{k} \left[\sum_{\mathrm{i}=1}^{\mathrm{N}} 1\left(\arg\max_{\mathrm{c}} p_c(\mathrm{x_I}) = k\right)\right], \quad (1)$$

where x_i is a target view, $p_c(\cdot)$ is the corresponding class c probability returned by the base classifier, $1(\cdot)$ denotes the indicator function, X is a set of one target’s views, $f_{mv}(X)$ is returns the class of the target.

The second approach defines the target class as (2):

$$f_{hp}(X) = \arg\max_{c} \left(\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} p_{\mathrm{c}}(\mathrm{x_i})\right), \quad (2)$$

where $f_{hp}(X)$ is a return the class of the target.

The third, Deep Sets-based classifier implements a function of the form (3):

$$f_{ds}(X) = \rho\left(\frac{1}{N}\sum_{i=1}^{N} \phi(x_i)\right), \quad (3)$$

where: $\phi(\cdot)$ is a shared feature extractor implemented using the model backbone, $\rho(\cdot)$ is a classification head producing target-level predictions and $f_{ds}(X)$ is the resulting class of the target [4].

3.2. Deep Models’ Architecture

All three approaches leverage transfer learning by reusing the backbone weights of an existing CNN-based object detector. The backbone is reused to leverage task-specific sonar features learned during detection training.

In the ensembling approaches, the single-view classifier (hereafter referred to as the base classifier) is obtained by adding a global average pooling layer and a two-class linear classification head on top of the detector backbone. The backbone consists of a CSPDarknet-based convolutional neural network previously trained for object detection in SSS waterfall imagery and reused as a shared feature extractor. The base classifier is subsequently fine-tuned on the target patches dataset, and target-level predictions are obtained by aggregating the corresponding outputs according to (1) and (2) at inference time.

The Deep Sets model builds on top of the object detector backbone by adding embedding-level aggregation: feature vectors extracted from all patches belonging to the same target are combined using permutation-invariant mean pooling, and the resulting target-level representation is classified by a two-class linear head. The resulting model is trained on the patches dataset fully leveraging multi-view target labelling. The target-level predictions are obtained as outputs of the model given a variable-size set of target patches as input.

3.3. Dataset Generation

Experiments were conducted on real multi-mission SSS datasets, where the same physical targets were observed multiple times across different missions, passes and acquisition conditions.

The mission logs were first imported into SonarWiz software package [18] such that data from different missions in the same location were combined into a single project. This way the annotation process was supported by cross-reference and verification across multiple side-scan files.

Each target observation was manually annotated with a bounding box, assigned one of the classes: “mine”, “mine.floating”, “mine.rockan”, “mine.manta”, “mine.mk”, NOMBO, and an integer target identifier. When association of multiple observations with the same target was difficult or not relevant, null target identifier was assigned. In the context of an end-to-end ATR system, association of multiple detections together can be performed with clustering [19] or by other means.

Later, 64×64 pixel patches of target observations were extracted from annotated waterfall diagrams. The patch size was fixed for both small and large targets so that the target occupied a relatively small central portion of the patch, and a substantial surrounding context was preserved for model learning. In the patches dataset, the NOMBO-labelled detections were converted to NOMBO patches, and other detections received “mine” label.

3.4. Dataset Description

The object detection dataset contained 903 waterfall images with 1019 detections generated from 12 missions in three different geographical locations. The resulting patches dataset contained 804 64×64 pixel patches belonging to 47 physical targets. The number of views per target varied from 1 to 70, reflecting realistic survey scenarios (Fig. 2).

As for class distribution, 442 patches belonged to “mine” class and 362 – to NOMBO class resulting in a relatively balanced split. Ground-truth class labels were provided at the patch and target level.

3.5. Training and Evaluation

For the training process, the number of patches was capped by 25 randomly selected images per target to avoid classification imbalance. Due to the limited number of samples, four-fold cross-validation was employed. The fold split was performed at the target level so that around one quarter of the targets appeared in the validation subset of each fold, honoring class balance, and patches of each target could only appear in either training or validation subset.

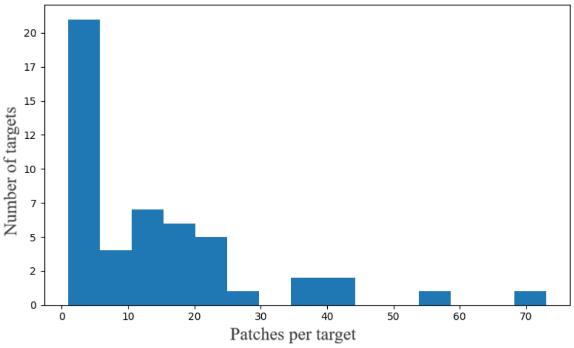


Fig. 2. Distribution of the number of patches per target.

The training was performed using Adam optimizer, learning rate of 0.0001, weight decay of 0.00001 for 1000 epochs, the best epoch was selected by the highest macro F1 score. All experiments were done using PyTorch [20] framework.

The performance was evaluated at patch level and target level (assuming that target level is the ultimate objective of real-world missions). For multi-view classifiers, the target-level class predictions were propagated to the corresponding patches for comparison. At both levels, standard classification metrics such as accuracy, precision, recall and F1-score were calculated. At the patch level, the base classifier served as the baseline, at the target level – the majority vote ensemble built upon the base classifier.

4. Results

4.1. Patch Level

Despite the moderate performance of the base classifier at patch level (68 % accuracy and 59 % F1 score), both the majority vote and the highest average probability ensembles showed a significant improvement after propagating target classifications to the corresponding patches (74 %, 78 % of accuracy and 61 %, 65 % of F1 score respectively), see Table 1. The Deep Sets-based approach achieved the strongest overall performance, with an accuracy of 90 % and an F1 score of 88 %, outperforming both ensemble strategies.

At the same time all three multi-view approaches demonstrated high recall for the mine class (95 %, 100 %, 100 %, correspondingly) compared to 82 % for the base classifier. This may provide substantial benefits for countermine operations and related applications. However, for both ensemble methods recall of NOMBO class dropped from 36 % in the base classifier to 27 % in majority vote and 28 % in highest average probability, that can be explained by the high appearance variability of NOMBO objects that ensembles could not treat well. At the same time Deep Sets kept recall for NOMBO relatively high (69 %).

Table 1. Classification performance metrics.
Note: Target-level metrics not available for base classifier as it operates on individual patches only.

Level (Entity)	Class	Metric	Base classifier	Majority vote	Highest average probability	Deep Sets
Patch	mine	Precision	0.74	0.74	0.76	0.88
		Recall	0.82	0.95	1.00	1.00
	NOMBO	Precision	0.46	0.70	1.00	1.00
		Recall	0.36	0.27	0.28	0.69
	Total	Accuracy	0.68	0.74	0.78	0.90
		F1 score	**0.59**	**0.61**	**0.65**	**0.88**
Target	mine	Precision	–	0.67	0.68	0.89
		Recall		0.96	1.00	1.00
	NOMBO	Precision		0.90	1.00	1.00
		Recall		0.45	0.45	0.86
	Total	Accuracy		0.72	0.74	0.94
		F1 score		**0.70**	**0.71**	**0.94**

4.2. Target Level

At the target level, Deep Sets achieved the highest accuracy and macro F1 score (both at 94 %), indicating consistent and reliable target-level predictions. It provided a significant increase from 72 % in accuracy and 70 % F1 score of the baseline majority vote method. All three methods kept high recall for the mine class (96 %, 100 %, 100 % correspondingly).

Within the ensemble methods, the highest average probability demonstrated slightly better performance in both classes, from one to ten percentage points higher than the baseline across the evaluated metrics.

Overall, the results show that all multi-view approaches improve performance compared to the baseline at both levels. Ensemble methods provide moderate gains, increasing accuracy and F1 score, particularly for the "mine" class through higher recall, however with reduced recall for the NOMBO class.

The Deep Sets-based approach achieves the best performance across all evaluated metrics, providing the highest accuracy and F1 scores and maintaining a more balanced classification between classes at both patch and target levels.

4.3. Performance Variability

Additional experiments using alternative target-level fold assignments and repeated runs produced a target-level F1 score of 91 ± 3 % for Deep Sets, compared to 70 ± 3 % and 71 ± 4 % for majority-vote and probability-based aggregation, respectively.

5. Discussion

The experimental results confirm that aggregating multiple observations consistently improves target classification in SSS imagery compared to traditional single-view approaches. Consequently, SSS target classification can be viewed as an inherently multi-view problem, and corresponding methods can be applied in this domain similarly to other sensing modalities.

Ensemble methods improve performance by combining predictions but assume equal contribution of views and do not learn a shared representation of a target. While probability-based ensemble outperforms majority vote by a few percentage points, showing the benefit of incorporating prediction confidence into the decision-making process, which is consistent with findings reported in [14, 15].

In contrast, the Deep Sets approach aggregates information at the embedding level and learns a target-level representation from multiple observations, providing higher and more balanced performance. This confirms the applicability of Deep Sets to multi-view sonar classification. The key property of the proposed approach is its ability to handle a variable number of views per target without architectural constraints, unlike some of the methods described in [12].

The provided dataset generation procedure enabled reliable and stable training and evaluation of deep models and can be recommended for target-aware SSS dataset generation.

From the evaluation perspective, patch-level metrics could be biased by the uneven number of patches per target, where misclassification of a single target with many patches can disproportionately affect overall results. In addition, the dataset aggregated multiple mine types (e.g. Manta, Rockan, Mk-36) into a single class, while the NOMBO class included diverse objects with highly variable appearance. This increased variability and caused the need for more representative and versatile datasets.

From a practical standpoint, the approach integrates naturally with existing detection pipelines by reusing the backbone of a CNN-based detector. However, it relies on target-level associations between detections, which may be affected by errors in localization, labeling and clustering in real conditions.

Future work includes exploring attention-based aggregation, jointly modeling detection and clustering, and extending the approach to multi-class and cross-domain classification with finer-grained taxonomy. Further validation on larger and more diverse datasets is required to assess generalization across different sonar platforms and conditions. Special attention needs to be paid to the distribution of the number of views per target and its influence on classification performance.

6. Conclusions

This paper addresses the problem of multi-view classification of SSS targets under realistic operational conditions. A framework for target-level classification based on aggregation of multiple detections was proposed and evaluated using real-world data. An approach for labeling, generating, and validating multi-view SSS datasets with explicit target associations was developed.

Three aggregation strategies were investigated, including decision-level ensemble methods and embedding-level aggregation using Deep Sets architecture. The results demonstrate that incorporating multi-view information significantly improves classification performance compared to single-view approaches. Among the evaluated methods, the model based on the Deep Sets approach provides the highest and most consistent performance. This makes it a practical solution for real-world applications, such as ATR in general and MCM in particular.

Acknowledgement

The authors thank EvoLogics GmbH for providing the equipment, data, and technical support. This research is a part of the OTC-Digital Twin & Analytics (OTC-DaTA) project (03ZU1107FB).

References

[1]. M. Motylinski, A. J. Plater, J. E. Higham, Computer vision methods for side scan sonar imagery, *Measurement Science and Technology*, Vol. 36, Issue 1, 2025, 015435.

[2]. A. Khan, M. M. Fouda, D.-T. Do, A. Almaleh, et al., Underwater target detection using deep learning: Methodologies, challenges, applications, and future evolution, *IEEE Access*, Vol. 12, 2024, pp. 12618-12635.

[3]. L. S. Gomes, et al., Sonar image datasets: A comprehensive survey of resources, challenges, and applications, in *Proceedings of the 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2025, pp. 1-6.

[4]. M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, et al., Deep sets, *arXiv*, 2018, arXiv:1703.06114.

[5]. L. Zhou, G. Du, K. Lü, L. Wang, et al., A survey and an empirical evaluation of multi-view clustering approaches, *ACM Computing Surveys*, Vol. 56, Issue 7, 2024, pp. 1-38.

[6]. Z. Chen, L. Fu, J. Yao, W. Guo, et al., Learnable graph convolutional network and feature fusion for multi-view learning, *Information Fusion*, Vol. 95, 2023, pp. 109-119.

[7]. Q. Teng, X. Yang, Q. Sun, P. Wang, et al., Sequential attention layer-wise fusion network for multi-view classification, *International Journal of Machine Learning and Cybernetics*, Vol. 15, 2024, pp. 5549-5561.

[8]. X. Tong, Y. Wang, Active multiview fusion framework for SAR automatic target recognition, *IEEE Transactions on Instrumentation and Measurement*, Vol. 73, 2024, pp. 1-14.

[9]. X. Liu, et al., Deep multiview union learning network for multisource image classification, *IEEE Transactions on Cybernetics*, Vol. 52, Issue 6, 2022, pp. 4534-4546.

[10]. J. Pei, Y. Huang, W. Huo, Y. Zhang, et al., SAR automatic target recognition based on multiview deep learning framework, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 56, Issue 4, 2018, pp. 2196-2210.

[11]. D. P. Williams, R. Hamon, I. D. Gerg, On the benefit of multiple representations with convolutional neural networks for improved target classification using sonar data, in *Proceedings of the Underwater Acoustics Conference and Exhibition (UACE)*, 2019, pp. 187-194.

[12]. B. d'Alès de Corbet, D. P. Williams, S. Dugelay, Target classification using multi-view synthetic aperture sonar imagery, in *Proceedings of the Underwater Acoustics Conference and Exhibition (UACE)*, 2019, pp. 227-233.

[13]. B. Zerr, B. Stage, A. Guerrero, Automatic target classification using multiple sidescan sonar images of different orientations, Technical Report SM-309, NATO SACLANT Undersea Research Centre, La Spezia, 1997, https://apps.dtic.mil/sti/pdfs/AD1120709.pdf

[14]. X. Wang, X. Liu, N. Japkowicz, S. Matwin, Automated approach to classification of mine-like objects using multiple-aspect sonar images, *Journal of Artificial Intelligence and Soft Computing Research*, Vol. 4, Issue 2, 2015, pp. 133-148.

[15]. S. Reed, Y. Petillot, J. Bell, An automatic approach to the detection and extraction of mine features in sidescan sonar, *IEEE Journal of Oceanic Engineering*, Vol. 28, Issue 1, 2003, pp. 90-105.

[16]. J. Fawcett, V. Myers, D. Hopkin, A. Crawford, et al., Multiaspect classification of sidescan sonar images: Four different approaches to fusing single-aspect information, *IEEE Journal of Oceanic Engineering*, Vol. 35, Issue 4, 2010, pp. 863-876.

[17]. Y. Steiniger, J. Groen, J. Stoppe, D. Kraus, et al., A study on modern deep learning detection algorithms for automatic target recognition in sidescan sonar images, in *Proceedings of the 6th Underwater Acoustics Conference and Exhibition (UACE)*, 2021, 070010.

[18]. Chesapeake Technology, SonarWiz sidescan, https://chesapeaketech.com/products/sonarwiz-sidescan/

[19]. O. Katrusha, Deep similarity model for improving clustering of sonar object detections, in Science, Technology and Industry in the Digital Age, *International Scientific Unity*, Hamburg, 2026, pp. 243-246.

[20]. J. Ansel, et al., PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation, in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2024, pp. 929-947.

(011)

Comparative Analysis of Deep Learning Models for UAV-Based Aerial Object Detections

A. Suyan, M. Hakami, Faleh Ali and S. Alhajaj
Prince Sattam bin Abdulaziz University, Collage of Electrical Engineering at Wadi-Aldawaser
Al-Kharj، Riyadh، Saudi Arabia
E-mail: s.alhajaj@psau.edu.sa

Summary: Unmanned Aerial Vehicles (UAVs) have become valuable platforms for aerial surveillance, monitoring, and object recognition. This study evaluates the performance of three deep learning architectures – Convolutional Neural Networks (CNN), EfficientNet, and ConvNeXt – for UAV-based aerial object detection using publicly available aerial image datasets. The models were trained and evaluated using the Adam and AdamW optimization algorithms on a representative subset of aerial vehicle images. Experimental results show that ConvNeXt combined with AdamW achieved the best performance, reaching a validation accuracy of 89 % and a validation loss of 0.072. EfficientNet with AdamW also delivered strong results, while the conventional CNN achieved slightly lower accuracy. The findings demonstrate the potential of modern deep learning architectures to enhance aerial object classification and support UAV applications such as surveillance, infrastructure monitoring, and autonomous inspection. This work provides a foundation for future research on real-time UAV-based object detection and search-and-rescue systems.

Keywords: UAV, Deep learning, Image classification, ConvNeXt, EfficientNet, CNN, Artificial intelligence.

---

1. Introduction

Recent advances in Unmanned Aerial Vehicles (UAVs) have expanded the use of aerial imaging systems in surveillance, transportation monitoring, infrastructure inspection, Search and Rescue, and environmental observation. While deep learning has achieved remarkable success in image classification tasks, its performance is often associated with the availability of large annotated datasets. In many practical applications, however, Artificial intelligence, particularly deep learning, has become a key technology for extracting meaningful information from aerial imagery and supporting real-time decision making.

1.1. Deep Learning Architectures

This study evaluates CNN, EfficientNet, and ConvNeXt architectures for aerial image classification tasks. CNN serves as a conventional benchmark architecture for image classification, learning hierarchical visual representations through convolutional feature extraction. [1]. EfficientNet is designed to improve learning efficiency through balanced network scaling [2], which might enhance performance when training data are limited. ConvNeXt incorporates modern architectural refinements that improve feature extraction and generalization capability [3]. Due to these differences, the three architectures are expected to exhibit varying levels of classification accuracy and robustness under limited-data conditions.

1.2. Optimization Algorithms

Optimization algorithms play an important role in the training and generalization of deep learning models. In this study, Adam and AdamW optimizers were selected due to their widespread use in image classification tasks. Adam provides adaptive learning rates for network parameters, enabling efficient convergence during training. AdamW extends Adam by decoupling weight decay from the optimization process, which can improve regularization and reduce overfitting. Evaluating both optimizers allows investigation of their impact on model performance under limited-data conditions.

1.3. Research Objective

This study presents a comparative evaluation of CNN, EfficientNet, and ConvNeXt architectures for aerial vehicle classification. The selected models are trained using Adam and AdamW optimization algorithms and assessed using aerial image datasets. The objective is to analyze their classification performance, generalization capability, and suitability for UAV-based image classification systems operating under limited-data and resource-constrained conditions.

2. Background

Recent advances in UAV technology have improved aerial monitoring, surveillance, and autonomous operation capabilities. Research has investigated energy-efficient UAV charging strategies [4], battery management and mission planning [5], cooperative multi-UAV exploration [6], and traffic monitoring systems [7]. In addition, UAV-based remote sensing has demonstrated significant potential in environmental monitoring and precision agriculture applications [8].

Artificial Intelligence (AI), particularly deep learning, has become a key technology for analyzing aerial imagery. Convolutional Neural Networks (CNNs) have been widely adopted for image classification and feature extraction tasks. More recent architectures such as EfficientNet improve learning efficiency through compound scaling strategies, while ConvNeXt incorporates modern architectural refinements to enhance feature representation and classification performance. These developments have enabled more accurate and robust aerial image classification systems, motivating the comparative evaluation presented in this study.

3. Experimental Framework

The experimental evaluation was conducted using an aerial vehicle classification dataset containing approximately 3200 images divided into vehicle and non-vehicle classes. Three deep learning architectures, namely CNN, EfficientNet, and ConvNeXt, were trained and evaluated using Adam and AdamW optimization algorithms under identical training conditions. All images were preprocessed and resized to meet the input requirements of the selected models. Each architecture–optimizer combination was trained for 100 epochs, resulting in six experimental configurations. Model performance was assessed using validation accuracy and validation loss to evaluate classification capability and generalization performance.

4. Results

Fig. 1 illustrates the training and validation accuracy curves of the CNN model trained using the Adam optimizer. The model demonstrates a steady improvement in classification performance throughout the training process, with both curves exhibiting similar trends. The relatively small gap between training and validation accuracy indicates acceptable generalization capability, while the final validation accuracy of approximately 82 % confirms the effectiveness of the CNN architecture for aerial image classification.

The performance of the CNN model using the AdamW optimizer is illustrated in Fig. 2. Compared with Adam, AdamW demonstrates improved convergence behavior and achieves higher validation accuracy, indicating better generalization on the evaluated dataset.

Fig. 3 presents the accuracy curves of EfficientNet trained with the Adam optimizer. The model initially exhibits a training accuracy of approximately 42 %, below the 50 % accuracy associated with random guessing in a balanced binary classification problem.

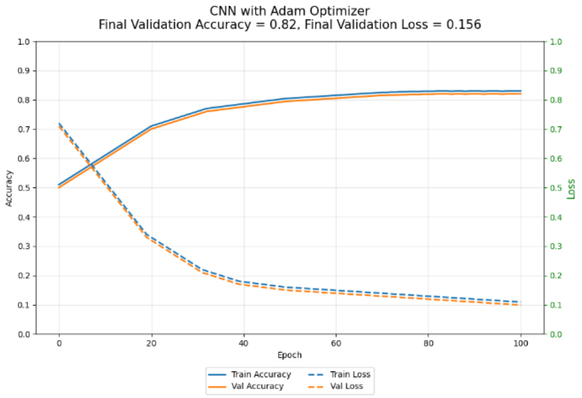


Fig. 1. CNN accuracy curves with Adam.

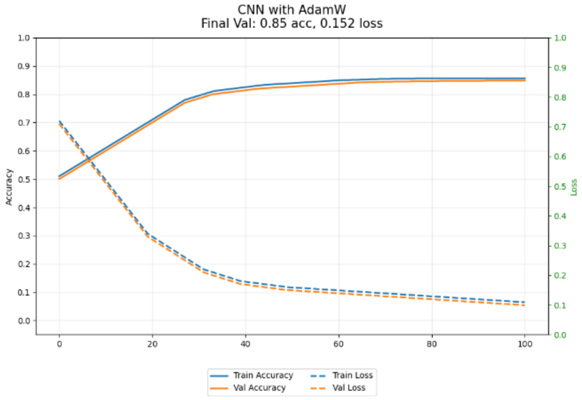


Fig. 2. CNN accuracy curves with AdamW.

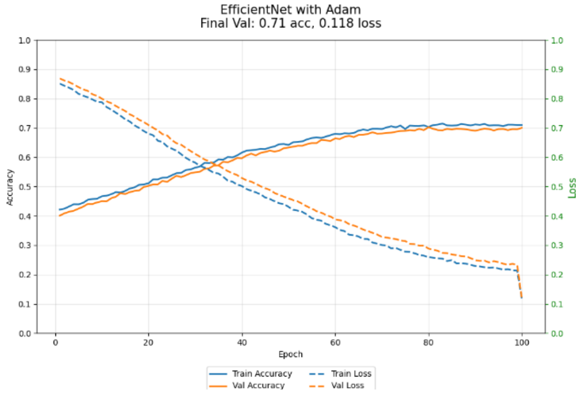


Fig. 3. EfficentNet accuracy curves with Adam.

As shown in Fig. 4, EfficientNet trained with the AdamW optimizer exhibits stable convergence and improved validation performance compared with the Adam-based model. The results suggest that AdamW enhances the model's generalization capability.

Fig. 5 presents the training and validation accuracy curves of the ConvNeXt model trained with the Adam optimizer. Similar to the EfficientNet model shown in Fig. 3, the training accuracy initially starts at approximately 42 %, which is below the 50 % accuracy expected from random guessing in a balanced binary classification task. Despite this behavior during the early training stages, the model later converges and achieves high validation accuracy.

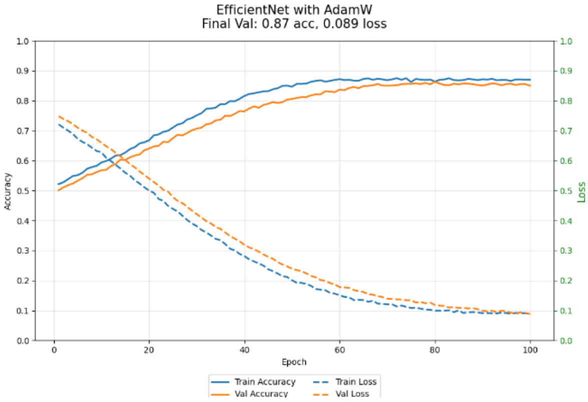


Fig. 4. EfficentNet accuracy curves with AdamW.

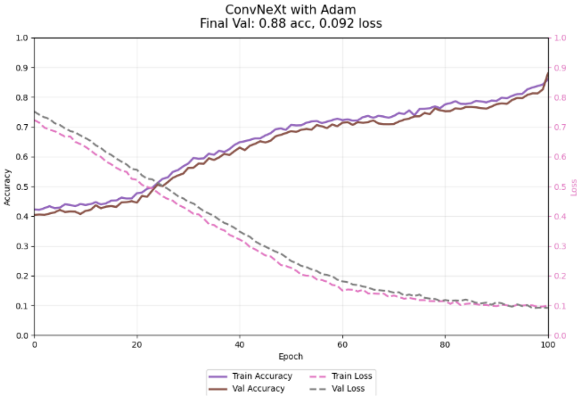


Fig. 5. ConvNeXt accuracy curves with Adam.

As shown in Fig. 6, ConvNeXt trained with the AdamW optimizer achieves the best overall performance among the evaluated models. The model exhibits stable convergence and strong agreement between training and validation accuracy, reaching a final validation accuracy of approximately 89 %.

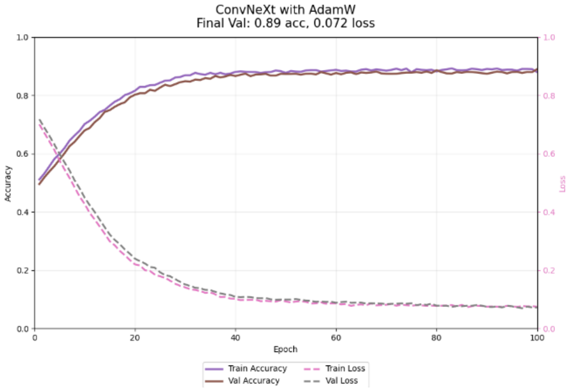


Fig. 6. ConvNeXt accuracy curves with AdamW.

Table 1 List all comparisons validation accuracies and validation losses done in this study.

Furthermore, Fig. 7 shows the training accuracy curves during the 100-epoch training process, illustrating how the performance of each model improves as learning progresses and eventually converges toward maximum accuracy.

Table 1. Comparison of accuracy and loss.

Model	Optimizer	Validation Accuracy	Validation Loss
EfficientNet	Adam	0.42	0.118
ConvNeXt	Adam	0.88	0.092
CNN	AdamW	0.85	0.152
CNN	Adam	0.82	0.156
ConvNeXt	AdamW	0.89	0.072
EfficientNet	AdamW	0.87	0.089

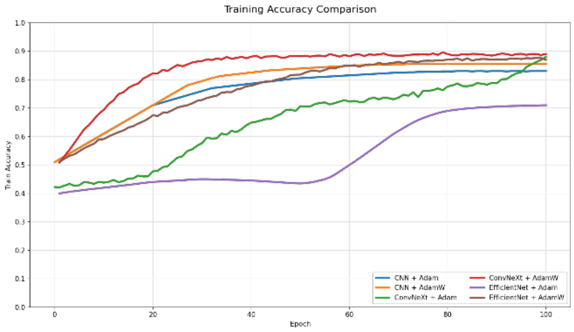


Fig. 7. All Training Accuracy Comparison.

5. Results Theoretical Analysis

To better understand the performance differences observed in the experimental results, the fundamental mathematical principles of the evaluated architectures are presented in this section. The following equations describe the core operations.

5.1. CNN

The convolution operation extracts local spatial features from the input image and forms the foundation of CNN feature learning [9].

The basic convolution operation is defined as:

$$y(i,j) = \sum_m \sum_n x(i+m, j+n).w(m,n) \quad (1)$$

And ReLU activation function:

$$f(x) = \max(0, x) \quad (2)$$

Softmax function for classification output:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}} \quad (3)$$

Max Pooling operation:

$$y(i,j) = max_{\substack{0 \le m < k \\ 0 \le n < k}} x(i+m, j+n) \quad (4)$$

5.2. EfficientNet

EfficientNet improves model performance through compound scaling, where the network depth (d), width (w), and input image resolution (r) are scaled uniformly according to [10]:

$$d = \alpha^{\phi},\ w = \beta^{\phi},\ r = \gamma^{\phi} \tag{5}$$

while the constraint:

$$\alpha . \beta^2 . \gamma^2 \approx 2 , \alpha \geq 1, \beta \geq 1 , \gamma \geq 1 \tag{6}$$

5.3. ConvNeXt

ConvNeXt uses depthwise separable convolution, [11] where each channel is processed independently:

$$y_c(i,j) = \sum_{m,n} x_c(i+m, j+n) . w_c(m,n) \tag{7}$$

The mathematical formulations presented in this section provide insight into the design principles of CNN, EfficientNet, and ConvNeXt. These architectural differences contribute to the performance variations observed in the experimental evaluation.

Layer Normalization (LN) applied across channels

$$LN(x) = \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} \times \gamma + \beta, \tag{8}$$

where

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i, \sigma^2 = \frac{1}{d}\sum_{i=1}^{d} (x_i - \mu)^2 , \tag{9}$$

6. Conclusion

This study evaluated the performance of CNN, EfficientNet, and ConvNeXt for aerial image classification using a limited dataset of approximately 3200 images. Each architecture was trained using both Adam and AdamW optimizers under identical experimental conditions. The results demonstrated that ConvNeXt combined with AdamW achieved the highest validation accuracy, indicating superior feature extraction and generalization capability compared to the other evaluated configurations.

Several notable observations were identified during the experiments. While most models exhibited expected learning behavior, both EfficientNet and ConvNeXt trained with the Adam optimizer initially achieved training accuracies of approximately 42 %, which is lower than the 50 % accuracy associated with random guessing in a balanced binary classification task. Despite this unusual starting point, both models successfully converged and achieved strong classification performance. In contrast, the corresponding AdamW configurations exhibited more stable learning behavior and consistently higher validation accuracy.

Overall, the findings suggest that modern architectures such as ConvNeXt can maintain robust performance even under limited-data conditions. Furthermore, the results indicate that the choice of optimizer can significantly influence convergence behavior and final classification performance, with AdamW providing the most consistent results across the evaluated architectures.

References

[1]. Y. LeCun, et al., Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, Vol. 86, Issue 11, 1998, pp. 2278-2324.

[2]. M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 6105-6114.

[3]. Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, et al., A ConvNet for the 2020s, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11976-11986.

[4]. H. Yan, Y. Chen, S. Yang, UAV-enabled wireless power transfer with base station charging and UAV power consumption, *IEEE Transactions on Vehicular Technology*, Vol. 69, Issue 11, 2020, pp. 12883-12896.

[5]. S. Jiao, G. Zhang, M. Zhou, G. Li, A comprehensive review of research hotspots on battery management systems for UAVs, *IEEE Access*, Vol. 11, 2023, pp. 84636-84650.

[6]. K. Cesare, R. Skeele, S. Yoo, Y. Zhang, et al., Multi-UAV exploration with limited communication and battery, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2230-2235.

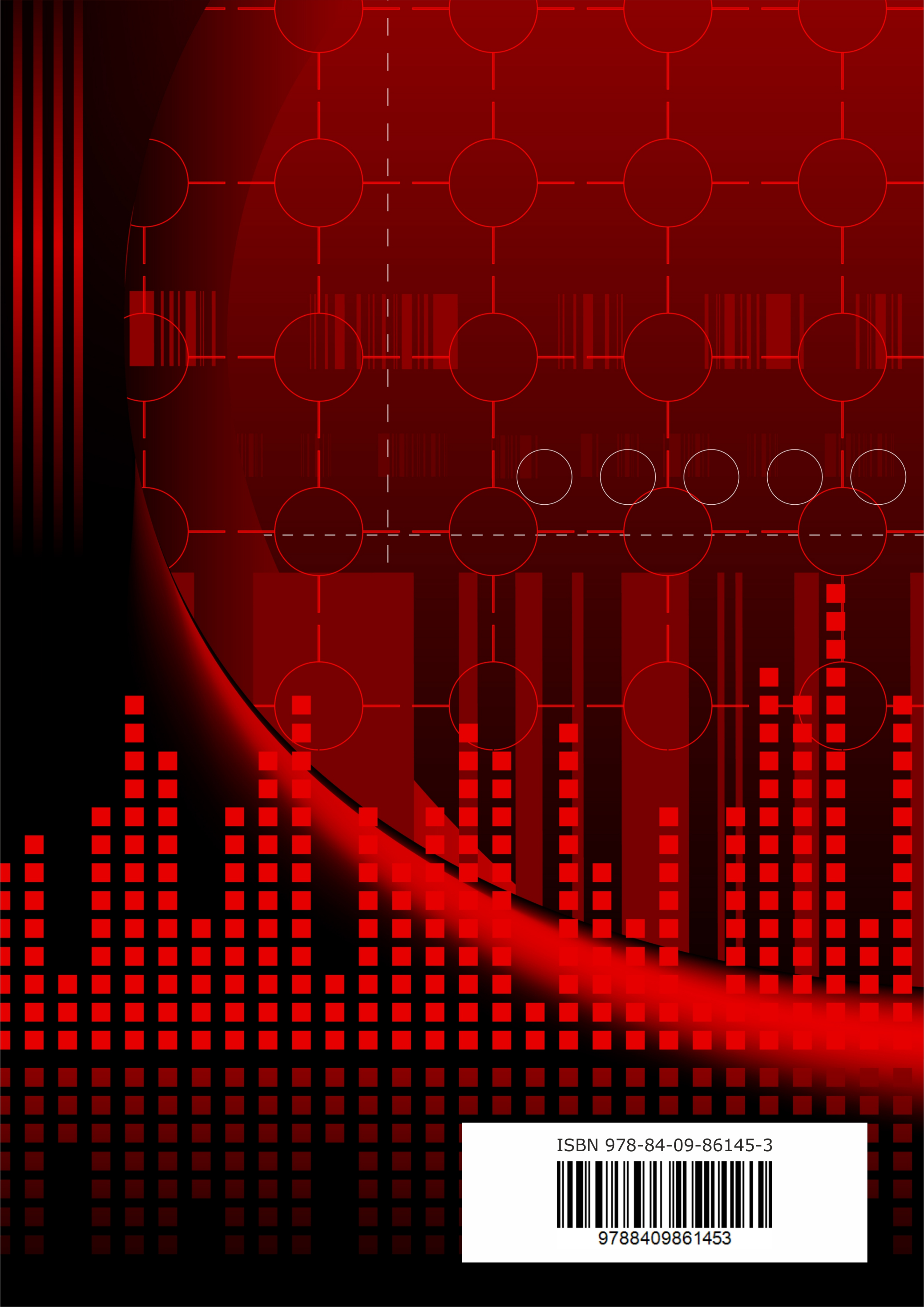
[7]. M. Elloumi, R. Dhaou, B. Escrig, H. Idoudi, et al., Monitoring road traffic with a UAV-based system, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1-6.

[8]. N. Yin, R. Liu, B. Zeng, N. Liu, A review: UAV-based remote sensing, *IOP Conference Series: Materials Science and Engineering*, Vol. 490, Issue 6, 2019, 062014.

[9]. I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, *MIT Press*, Cambridge, 2016.

[10]. D. Foster, Generative Deep Learning, *O'Reilly Media*, Sebastopol, 2023.

[11]. A. Howard, M. Zhu, B. Chen, D. Kalenichenko, et al., MobileNets: Efficient convolutional neural networks for mobile vision applications, *arXiv*, 2017, arXiv:1704.04861.



ISBN 978-84-09-86145-3

9788409861453