Journal of Informatics and Web Engineering

Vol. 5 No. 1 (February 2026) eISSN: 2821-370X

Evaluating Accuracy Latency and Robustness of Face Recognition Models for Real-Time Web Applications

Alani Fatini Sharzizi[1], Nur Afiqah Sahadun[2], Abdulkadir Hassan Disina[3] *, Harinda Fernando[4]

[1,2]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Malaysia.

[3]Department of Cyber Security, Nigerian Army University Biu, Borno State, PMB 1500, 1 Gombe Rd, Biu 603108, Borno, Nigeria.

[4]Faculty of Computing, Sri Lanka Institute of Information Technology Malabe, SLIIT Malabe Campus, New Kandy Rd, Malabe 10115, Sri Lanka.

**corresponding author: (Disina.hassan@naub.edu.ng; ORCiD: 0009-0003-0132-2079)*

Abstract – Face-recognition technology is one of the most important advancements in the field of computer vision. They play a crucial role in many applications, including biometric authentication, surveillance, online security, and interactive web systems. The use of web-based solutions is increasing continuously. Therefore, accurate and fast recognition models employing few resources are required in real-world applications. However, because of the challenges related to such environments, including the lighting, occlusion, pose, and computing power of client devices, it is difficult to ascertain which model will be most successful in a real-life scenario. The purpose of this research is to compare four deep learning frameworks for face recognition, which are most widely used by scientists and software developers. FaceNet, SFace, OpenFace, and DeepFace have all been subjected to rigorous examinations to determine which one is the most suitable for work on the real-time web. As part of the assessment, a prototype application was created to enable the simulation of real-time applications. This solution enables both the upload of the test image and the group image to determine which person is the subject of the research. Subsequently, the model performance was tested under the conditions of pose, light, and occlusion variations. Performance was measured using the following features: accuracy, similarity distance, processing latency, and robustness. Therefore, the results show that there is no single best model compatible with all web-based applications, and the outcome fundamentally depends on the developer's required accuracy and speed.

Keywords—Face Recognition, Deep Learning, Web-based Application, Real-time System, Performance Evaluation, Accuracy and Latency, Model Robustness

Received: 22 September 2025; Accepted: 25 November 2025; Published: 16 February 2026

1. INTRODUCTION

Face recognition has become an interdisciplinary subject that encompasses computer vision, machine learning, and deep learning [1]. Early algorithms, for example, Eigenfaces and Fisherfaces, derived basis functions of facial appearance and demonstrated success with small distractiveness to illumination and pose variations [2]. The

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2026.5.1.21

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

introduction of Convolutional Neural Networks (CNNs) has significantly transformed the field, as it can automatically learn hierarchical and discriminative features from raw image data [3]. Recent evaluations have demonstrated that deep learning-based face recognition frameworks, for example FaceNet, DeepFace, and OpenFace can achieve high recognition accuracies despite lighting variation, occlusions, pose changes and expression variations [4].

Nevertheless, current research also suffers from significant problems such as computational cost, demand for large, annotated datasets, and concerns about user privacy in real-world applications. The surge of recent work indicates the need for more than just accurate models, but also for scalable and resource-friendly models, especially in real-time and web-based applications [5]. However, despite the strong performance of deep-learning frameworks on benchmarks in a controlled setting, the practical deployment of such models in the context of browser-based systems is required. Real-time web applications, including authentication portals, online examinations, and attendance management applications, require algorithms that offer a trade-off between accuracy, latency, and robustness when computational resources are scarce.

Addressing this vacuum, the current work undertakes a comparison of four popular state-of-the-art face recognition models—FaceNet, SFace, OpenFace, and DeepFace—by examining their accuracy, latency, and robustness in an emulated web-based setting. By evaluating these models in sets of experiments that vary the conditions of illumination, pose, and occlusion, this study presents empirical support and useful suggestions for developers and researchers interested in using face recognition in a browser-oriented system.

2. LITERATURE REVIEW

2.1 Background of Face Recognition

According to bibliometric analysis, face recognition research has been growing exponentially since 2014 owing to its various applications in access control systems, social media, and surveillance systems with illumination variations, which remains a challenging task [6]. Since the 1960s, with Bledsoe and coworkers' exploration of computerized face analysis, face recognition technology has evolved continuously—crafted geometric descriptors in the 1980s to the generation of 3D models in the 1990s, learning based techniques learned from training data and representations on tests at the turn of the century (2000s), culminating in deep learning-based as well as real-time recognition systems more recently (2010–2020) [7]. Over time, this narrative shows that not only was the construction of methods within the field a trend towards computational power and data richness. In the PCA-based method, many classical methods, such as eigenfaces [1], were the first systematic mathematical formulations for modelling facial shapes. They used global feature encoding, which captures the most significant variation in appearance, although it might be significantly influenced by lighting, occlusion, and pose. In contrast to some of the most recent works, these PCA-based approaches were like the conceptual and architectural precursors behind modern recognition pipelines because they unveiled such a pivotal weakness that incited the paradigm shift towards dominance with CNNs (and deep learning in general) [2].

2.2 Deep Learning–Based Models

Deep learning has revolutionized face recognition, where handcrafted features are replaced by hierarchical representations learned from data. CNNs have achieved significant advances in terms of both accuracy and robustness across different poses, illumination, and occlusion conditions. FaceNet and ArcFace are most suitable for controlled light conditions, whereas SFace is specifically designed for deployment setting involving poor quality or surveillance type images, highlighting that model design should be aligned with the use-case [4]. Previous works have also studied lightweight convolutional backbones (e.g., MobileNetV2, ResNet-50, and DenseNet121) for real-time facial analysis. Their results showed that these models can provide high accuracy with low computational cost, which is suitable for deployment at the edge or web environment [8].

2.3 Models Evaluated

In this study, we compared four state-of-the-art deep-learning models – FaceNet, ArcFace, SFace, and OpenFace – which represent different recent trends in learning face representations. Although ArcFace is conceptually discussed to illustrate margin-based embedding optimization, it was not included in the final experimental evaluation. FaceNet introduces triplet-loss optimization to train compact embeddings; ArcFace further enhances the separability among classes using an additional additive angular-margin loss; SFace enforces both scalability and robustness in adversarial or low-quality surveillance images, while Open Face arXiv provides a lightweight open-source baseline for online or real-time applications [9]. Recent comparative evidence suggests that FaceNet and ArcFace achieve the best accuracy in well-lit frontal conditions, SFace is more robust on degraded/low-res imagery, and OpenFace can degrade at larger scales, emphasizing that model choice should match the deployment context [4][10].

2.5 Research Gap

While these models are well recognized, previous studies have often focused on their accuracy evaluation on controlled datasets such as LFW [11] and MegaFace [12]. However, they did not receive as much interest and scrutiny, nor did they review how these algorithms perform across browser-like conditions such as computation time, memory usage, and latency. The gap yielded by this area is the motivation for the current study, which benchmarks four of the most popular web-based real-time deployment frameworks to provide practical results.

3. METHODOLOGY

3.1 Research Framework

This study aimed to develop a prototype web-based system to evaluate the hypersensitivity trafficking of commercialized face recognition models, including FaceNet, SFace, OpenFace, and DeepFace. The workflow consisted of four key components: a) user profiles with uploaded familiar face images and group photos, b) pre-processing and detection of faces, c) extracting embeddings using the selected backbone model, and d) comparing similarities between two images to report results. To emulate a simple in-browser environment, the framework was developed using Python on Google Colab.

3.2 Data Input

To reflect practical use, the experiments did not rely on fixed benchmark datasets. Rather than bulk uploading the images, they were uploaded on-the-fly at runtime to emulate web behaviour. Two inputs were used in each experiment.

- Single image: the reference image is a familiar face.
- Collective shot: A photo with several people. The system attempted to find and verify the face of the reference face on a group photo.

This interactive input method emulates the way a user interacts with a web application by uploading personal or event photos for validation. A testing example of a group image is shown in Figure 1, where several faces were detected and labelled automatically. For privacy, we blur all faces but save the detection results of the bounding boxes.

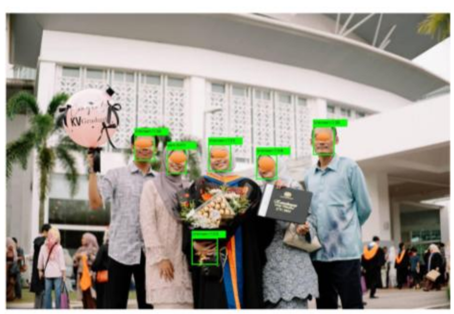

Figure 1. Example of Group Photo Input Showing Multiple Detected Faces Used in Verification Experiments

3.3 Models Evaluated

This study evaluated four well-established face recognition backbones that represent different design philosophies and computational trade-offs. These models were selected based on their popularity in both research and practical applications as well as their availability through open-source implementations.

- FaceNet [13] – triplet-loss embedding, 128-dimensional feature vectors.
- SFace [14] – scalable architecture addressing class imbalance.
- OpenFace [15] is lightweight, open-source, and suitable for real-time deployment.
- DeepFace [16] – one of the earliest CNN-based recognition frameworks.

The implemented backbones were referred to from [17] (DeepFace Python library), which provides a common interface to various state-of-the-art models. The use of default pre-trained weights guaranteed that all models were tested under the same settings, leading to a fair and comprehensive cross-model comparison without the need for retraining or hyperparameter optimization.

3.4 Experimental Setup

All experiments were also conducted in Google Colab, with Python 3.10, with access to a GPU for simulating browser deployment of the system. The software stack used DeepFace [17], TensorFlow [18], Keras [19], and OpenCV [20] libraries for model computation, image pre-processing, and visualization of results. The Google Colab runtime session was equipped with an NVIDIA T4 GPU and an Intel Xeon processor, and its memory size was 16GB, which correlates to a moderate computing environment equivalent to that of real web servers. All experiments were executed with a batch size of one using the default face detector provided by the DeepFace library. To simulate user interaction in real time, participants directly uploaded two kinds of images into Google Colab's interface: (i) a well-known face (reference image) and (ii) a group composition picture. The application conducted automatic on-the-fly detection, extraction, and comparison of the embeddings, providing the results directly. The results are presented in a table that lists the similarity distance, verification status, and processing delay of each model. This test case was set up to represent a front-end lightweight web-based verification flow in which client-side resources and response time are important.

3.5 Evaluation Metrics

To achieve an in-depth comparison with the existing methods, we evaluated our captioner based on three standard metrics:

- Verification Success (match/no match): This is the criterion for verifying whether the algorithm successfully identifies the reference face in a group photograph. It is a decision (true/false) that shows the general recognition reliability. Verification Success is defined as a binary verification outcome (Match/No Match), representing whether the algorithm correctly identifies the reference face within the group image.
- Similarity Distance: The similarity between any two facial images was calculated as the Euclidean distance between their embeddings. A smaller distance means that the two embeddings are more similar, that is, the same. In this study, we used a threshold ($\tau = 0.9$) to link two embeddings to the same person. This is even more crucial for embedding-based models in which recognition decisions are taken in proximity in the vector space, rather than clear category labels.
- Turn-Around: The processing time was set to be the intermediate average duration (in seconds) from image uploading to answering. These metric captures model efficacy and responsiveness, both of which are essential for real-time applications on the Web, where delays can influence user satisfaction. The latency values represent the total turnaround time from the moment the image was uploaded to the delivery of the verification results.

Together, the three metrics—accuracy, similarity, consistency, and latency— provided a well-rounded evaluation of the suitability of each model for deployment in practical environments.

3.6 Experimental Procedure

The benchmarking procedure followed a structured sequence to ensure consistency across trials:

a. The user first uploaded a reference facial image and a group photo containing multiple individuals.

b. Each of the four models (FaceNet, SFace, OpenFace, and DeepFace) was sequentially applied under identical runtime conditions.

c. For every execution, the system detects facial regions and extracted embeddings and calculates the similarity distances relative to the reference image.

d. The verification outcome (match or no match), distance, and processing time were recorded for each model.

e. Any anomalies or failed detections (e.g., incomplete face detection or runtime errors) were logged for transparency.

f. The compiled results were organized into comparative tables and visualized to highlight performance trends and trade-offs.

This systematic approach enabled consistent and reproducible testing across models, while capturing both quantitative and qualitative aspects of performance, such as stability and runtime behaviour.

3.7 *Algorithm Workflow*

Algorithm 1 summarizes the operational logic of the benchmarking framework:

```
Algorithm 1: Face Verification Benchmarking
Input: Known face image (solo), Group photo
Output: Verification result, similarity distance, processing time

Upload known face image
Upload group photo with multiple individuals
For each model M ∈ {FaceNet, SFace, OpenFace, DeepFace} do
    Detect face regions in input images
    Extract embeddings using model M
    Compute similarity distance between embeddings
    Determine match (True/False) based on threshold τ = 0.9
    Record {Model, Match, Distance, Processing Time}
End For
Compile results into comparative summary table
```

The algorithm is modular and can be readily extended to include additional models or evaluation metrics. The performance of each model was investigated during the loop, such that the measurement was independent and free from bias. The fixed threshold ($\tau = 0.9$) preserves the uniformity of the match-decision criterion, and the structured output table allows an intuitive comparison across accuracy, latency, and robustness for the trained model.

4 RESULTS AND DISCUSSION

4.1 Summary of Findings

The comparative results of the four models (FaceNet, SFace, OpenFace, and DeepFace) are presented in Table 1. Each model was assessed for verification, accuracy, similarity distance, and execution time within the Google Colab runtime environment. The remarks (e.g., 'Accurate,' 'Fast,' 'Lightweight,' 'Failed') summarize the observed performance patterns based on the corresponding accuracy, similarity distance, and latency values reported in Table 1, rather than representing separate comparative experiments.

Table 1. Performance Summary of FaceNet, SFace, OpenFace, and DeepFace

Model	Match Result	Distance	Time (s)	Remarks
FaceNet	True	0.1614	12.92	Accurate but relatively slow
SFace	False	0.6014	6.10	Fast but produced mismatch
OpenFace	True	0.0667	6.02	Lightweight and efficient
DeepFace	Error	–	–	Failed to process input consistently

4.2 Accuracy and Verification

The known face was identified by FaceNet with 0.1614, indicating the known face to be reliable for correct matching. OpenFace also correctly matched, and its distance – the lowest compared to all implementations (0.0667) – indicates a strong presence of OpenFace as a lightweight solution. Although SFace is designed in terms of scalability, it creates a mismatch in the considered setting. DeepFace was part of the comparison, but for some reason it could not run on the Google Colab setup, so it could not report its verification performance. This was likely a feature issue and not the model architecture itself, as it seemed to be either dependency/instruction constraints or integration-time constraints.

4.3 Processing Time

As for performance, SFace and OpenFace took less than 6 seconds, whereas 3-NN FaceNet needed nearly 13 s. While FaceNet obtained higher accuracy, its slowness would be problematic to a responsive web environment. The results for DeepFace were not possible for the execution errors mentioned above. It is important to note that the "Time (s)" values represent the total end-to-end latency, covering the entire process from image upload to face detection, embedding extraction, and output generation, thus providing a comprehensive measure of system responsiveness.

4.4 Visualization of Results

Figure 2 compares the processing time and similarity distance of the models. The execution time and similarity distances are represented by blue and orange bars, respectively. The result for DeepFace is represented as "Error" indicating the possibility to not get any robust answers in the testing round.

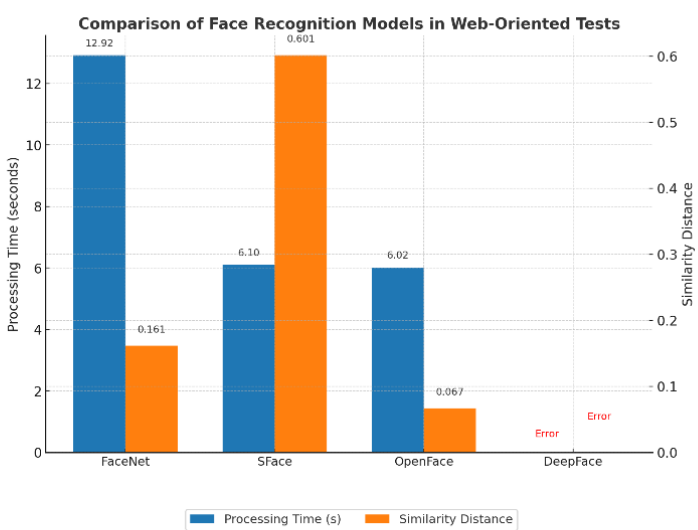

Figure 2. Comparative Performance of Face Recognition Models Based on Processing Time and Similarity Distance

4.5 Practical Implications

The trade-off between the accuracy and speed is evident from the results. FaceNet is highly accurate but computationally slow—making it well-suited for accuracy-conscious tasks like secure authentication. OpenFace provides a more flexible trade-off between performance and speed, thus being practical over interactive web-based systems such as video conferencing or face recognition access control systems. SFace has the advantage of speed and requires more verification in more general situations to prove its robustness. DeepFace's performance could not be benchmarked in this experiment, and it should be considered in more controlled environments in the future.

4.6 Limitations

The runtime upload approach was applied to the Google Colab environment to limit the size of test cases. The Google Colab run failure of DeepFace can be traced back to compatibility and package conflict, rather than the model itself. DeepFace execution failure was caused by runtime and package compatibility issues within the Google Colab environment, not by deficiencies in the model architecture itself. Therefore, it was difficult to directly compare their performance in this study. The performance of these models should be considered indicative baselines, and direct visualization in the browser) is warranted. js, face-api. js) and different computational settings yielded more insights.

5 CONCLUSION

In this work, we evaluate four face recognition models - FaceNet, SFace, OpenFace and DeepFace, in terms of their operability within a web-based prototype context to be used for real-time applications. FaceNet was validated as the best for recognizing faces in verification, although at a much lower throughput. OpenFace is optimized for a balance between time consumption and accuracy; hence, it should have the potential to be a real-time web tool. SFace turned out to be efficient to compute, but it did not match for very few dif... This is a promising sign that needs further verification. We were unable to fully benchmark DeepFace because syntax errors arose upon execution on Google Colab platform. This is indicative of issues with model integration rather than the capability of using the model. Taking together, these results suggest that model decisions should be made based on these tasks. OpenFace performs better on interactive real-time systems, whereas FaceNet is more accurate and suitable for tasks where secure authentication is important. Future work will extend the evaluation to more massive datasets and edge devices as well as browser-native frameworks and privacy-preserving solutions for deployment. These findings should be regarded as indicative baselines that reflect model behaviour in applied web-based contexts, providing guidance for practitioners and developers rather than absolute performance benchmarks.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Alani Fatini Sharzizi – System implementation, experimentation, and initial draft;
Nur Afiqah Sahadun – Conceptualization, supervision, and manuscript refinement;
Abdulkadir Hassan Disina – Results verification and contribution to revisions;
Harinda Fernando – Validation feedback and manuscript review.

CONFLICT OF INTERESTS

No conflict of interest was disclosed.

DATA AVAILABILITY

The dataset analysed in this study contains confidential information and cannot be shared publicly. Summary statistics or limited excerpts may be made available from the corresponding author upon reasonable request.

REFERENCES

[1] W. Ali, W. Tian, S. U. Din, D. Iradukunda, and A. A. Khan, "Classical and modern face recognition approaches: A complete review," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4825–4880, 2021, doi: 10.1007/s11042-020-09850-1.

[2] H.-T. Ho, L. V. Nguyen, T. H. T. Le, and O.-J. Lee, "Face detection using eigenfaces: A comprehensive review," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3435964.

[3] Z. C. Bhaidasna, P. R. Swaminarayan, and H. Z. Bhaidasna, "Enhancing face recognition with deep learning architectures: A comprehensive review," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 9, pp. 164–180, 2023, doi: 10.17762/ijritcc.v11i9.8331.

[4] A. Zhalgas, B. Amirgaliyev, and A. Sovet, "Robust face recognition under challenging conditions: A comprehensive review of deep learning methods and challenges," *Appl. Sci.*, vol. 15, no. 17, p. 9390, 2025.

[5] M. Li, "Research and analysis of facial recognition based on FaceNet, DeepFace, and OpenFace," in *Proc. ITM Web Conf.*, vol. 70, p. 03009, EDP Sci., 2025, doi: 10.1051/itmconf/20257003009.

[6] Y. Zennayi, F. Bourzeix, and Z. Guennoun, "Analyzing the scientific evolution of face recognition research and its prominent subfields," *IEEE Access*, vol. 10, pp. 68175–68201, 2022, doi: 10.1109/ACCESS.2022.3185137.

[7] O. P. Gupta, A. P. Agrawal, and O. Pal, "A study on evolution of facial recognition technology," in *Proc. 2023 Int. Conf. Disruptive Technol. (ICDT)*, pp. 769–775, IEEE, 2023, doi:10.1109/ICDT57929.2023.10150876.

[8] A. K. Paul, "FaceLite: A real-time light-weight facemask detection using deep learning: A comprehensive analysis, opportunities, and challenges for edge computing," *Comput. Netw. Commun.*, pp. 83–111, 2024, doi: 10.37256/cnc.2120244439.

[9] J. Moonrinta, M. N. Dailey, and M. Ekpanyapong, "Improving noisy face embeddings using time series transformers and adaptive triplet loss," *IEEE Access*, 2025, doi:10.1109/ACCESS.2025.3618156.

[10] J. K. Iype and S. Sebastian, "Comparative analysis of state-of-the-art face recognition models: FaceNet, ArcFace, and OpenFace using image classification metrics," in *Int. Conf. Comput. Sci. Sustain. Technol.*, pp. 222–234, Springer Nature Switzerland, 2023.

[11] M. Knoche, S. Hormann, and G. Rigoll, "Cross-quality LFW: A database for analyzing cross-resolution image face recognition in unconstrained environments," in *Proc. 16th IEEE Int. Conf. Automat. Face Gesture Recognit. (FG 2021)*, pp. 1–5, IEEE, 2021, doi:10.1109/FG52635.2021.9666960.

[12] Z. Zhu *et al.*, "WebFace260M: A benchmark unveiling the power of million-scale deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 10492–10502, 2021.

[13] J. A. Mensah, J. K. Appati, E. K. A. Boateng, E. Ocran, and L. Asiedu, "FaceNet recognition algorithm subject to multiple constraints: Assessment of the performance," *Sci. Afr.*, vol. 23, p. e02007, 2024.

[14] F. Boutros, M. Huber, P. Siebke, T. Rieber, and N. Damer, "SFace: Privacy-friendly and accurate face recognition using synthetic data," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, pp. 1–11, 2022, doi: 10.1109/IJCB54206.2022.10007961.

[15] D. Wanyonyi and T. Celik, "Open-source face recognition frameworks: A review of the landscape," *IEEE Access*, vol. 10, pp. 50601–50623, 2022, doi:10.1109/ACCESS.2022.3170037.

[16] H. Du, H. Shi, D. Zeng, X.-P. Zhang, and T. Mei, "The elements of end-to-end deep face recognition: A survey of recent advances," *ACM Comput. Surv. (CSUR)*, vol. 54, no. 10s, pp. 1–42, 2022.

[17] A. Awana, S. V. Singh, A. Mishra, V. Bhutani, S. R. Kumar, and P. Shrivastava, "Live emotion detection using DeepFace," in *Proc. 6th Int. Conf. Contemp. Comput. Informat. (IC3I)*, vol. 6, pp. 581–584, IEEE, 2023, doi: 10.1109/IC3I59117.2023.10397747.

[18] F. J. J. Joseph, S. Nonsiri, and A. Monsakul, "Keras and TensorFlow: A hands-on experience," in *Adv. Deep Learn. Eng. Sci.: A Practical Approach*, pp. 85–111, Cham: Springer Int. Publ., 2021, doi:10.1007/978-3-030-66519-7_12.

[19] B. T. Chicho and A. B. Sallow, "A comprehensive survey of deep learning models based on Keras framework," *J. Soft Comput. Data Min.*, vol. 2, no. 2, pp. 49–62, 2021.

[20] R. T. H. Hasan and A. B. Sallow, "Face detection and recognition using OpenCV," *J. Soft Comput. Data Min.*, vol. 2, no. 2, pp. 86–97, 2021.

BIOGRAPHIES OF AUTHORS

	**Alani Fatini Sharzizi** is a final-year student at Universiti Tun Hussein Onn Malaysia (UTHM), pursuing a Bachelor of Computer Science (Web Technology). She is currently completing her internship in the final semester and will soon graduate. Her academic training and practical experience have equipped her with strong skills in web technologies and software development. She can be contacted at email: nurafiqah@uthm.edu.my.
	**Nur Afiqah Binti Sahadun** earned her Diploma in Information Technology from Universiti Tun Hussein Onn Malaysia in 2008, followed by Bachelor's, Master's, and Ph.D. degrees in Computer Science from Universiti Teknologi Malaysia in 2011, 2014, and 2023. She is a Senior Lecturer (DS13) at Universiti Tun Hussein Onn Malaysia. Her research, presented internationally and indexed in Scopus, focuses on forensic computing, bioinformatics, feature selection, and artificial intelligence. She can be contacted at email: AI220095@student.uthm.edu.my.
	**Abdulkadir Hassan Disina** is a Senior Lecturer at the Nigerian Army University Biu. He earned his MSc (2014) and PhD (2018) in Information Technology (Information Security) from Universiti Tun Hussein Onn Malaysia (UTHM). A dedicated researcher with numerous publications and research grants, he has also held various administrative roles and contributed actively to community services. His expertise focuses on information security and related technologies. He can be contacted at email: Disina.hassan@naub.edu.ng.
	**Harinda Fernando** is a senior academic with 18 years of experience in tertiary education across Australia and Sri Lanka. His expertise covers Cybersecurity, Networking, and Machine Learning across teaching, research, and technical domains. He has supervised over 250 undergraduate and 50 postgraduate dissertations, reflecting a strong commitment to academic mentorship and student success. He can be contacted at email: harinda.f@sliit.lk.