ZONAsi Jurnal Sistem Informasi
ZONAsi VOL. 8 NO. 1 Jan 2026
Page: 96 - 110 Akreditasi Sinta 4
ISSN: 2656-7407 (Online)
2656-7393 (Print)

A HYBRID INTELLIGENCE FRAMEWORK FOR PERSONALIZED EYEWEAR RECOMMENDATION: INTEGRATING HAAR CASCADE LOCALIZATION WITH A GENETICALLY TUNED CNN CLASSIFIER

Abimanyu Putra Aria[1], Tri Widodo[2]
[1,2]Informatics Study Program, Faculty of Computer Science,
University of Technology Yogyakarta
St. Siliwangi, Jombor Lor, Sendangadi, Mlati, Sleman, D.I. Yogyakarta, Phone. (0274) 623310
e-mail: **[1]arraria3566@gmail.com**, [2]triwidodo@uty.ac.id

Abstract

A hybrid intelligence framework was presented to improve eyewear recommendations through robust face shape classification. A Haar Cascade Classifier was integrated for initial facial localization, alongside a Convolutional Neural Network (CNN) based on the InceptionV3 architecture for the primary classification task. A pseudo-labeling technique was utilized to refine the dataset, which elevated the initial model to 88.54% test accuracy. Furthermore, the CNN's hyperparameters were systematically tuned using a Genetic Algorithm (GA). This evolutionary tuning process yielded a significant performance boost, culminating in a final classification accuracy of 97.65%. It was concluded that the synergistic combination of advanced preprocessing, data refinement, and a genetically optimized deep learning model provided a highly accurate solution for personalized recommendation systems.
Keywords: *Face Shape Classification, Hybrid Intelligence, Convolutional Neural Network, Genetic Algorithm, Hyperparameter Optimization.*

Abstrak

Kerangka kerja kecerdasan hibrida disajikan untuk meningkatkan rekomendasi kacamata melalui klasifikasi bentuk wajah yang kuat. Klasifikasi Haar Cascade diintegrasikan untuk lokalisasi wajah awal, bersamaan dengan Jaringan Saraf Konvolusional (CNN) berbasis arsitektur InceptionV3 untuk tugas klasifikasi utama. Teknik pseudo-labeling digunakan untuk menyempurnakan dataset, yang meningkatkan akurasi pengujian model awal menjadi 88,54%. Selanjutnya, hyperparameter CNN disetel secara sistematis menggunakan Algoritma Genetika (GA). Proses penyetelan evolusioner ini menghasilkan peningkatan kinerja yang signifikan, yang berpuncak pada akurasi klasifikasi akhir sebesar 97,65%. Disimpulkan bahwa kombinasi sinergis dari pra-pemrosesan tingkat lanjut, penyempurnaan data, dan model pembelajaran mendalam yang dioptimalkan secara genetik memberikan solusi yang sangat akurat untuk sistem rekomendasi yang dipersonalisasi.
Kata kunci: *Klasifikasi Bentuk Wajah, Kecerdasan Hibrida, Jaringan Saraf Konvolusional, Algoritma Genetika, Optimasi Hiperparameter*

1. PRELIMINARY

The confluence of fashion and technology has amplified the importance of personalized accessory recommendation, with eyewear being a prominent category. The selection of an appropriate spectacle frame is not merely a functional choice for vision correction but also a significant aesthetic decision that can enhance or detract from an individual's facial harmony [1], [2]. The cornerstone of an effective eyewear recommendation system lies in its ability to accurately classify the user's face shape, as certain frame styles are known to complement specific facial geometries [3]. However, the manual classification of face shapes is subjective and inconsistent, while automated methods face significant challenges due to variations in pose, lighting, and expression in digital images [4], [5].

To address these challenges, the introduction section provides clear information about the background and problem formulation. This research proposes a hybrid intelligence framework that automates and optimizes the face shape classification process for personalized eyewear recommendation. The framework is built upon the synergistic integration of three key technologies: a robust face localization method, an advanced deep learning classifier, and a sophisticated optimization algorithm.

First, for the critical preprocessing step of face localization, the *Haar Cascade Classifier* is employed. This method is renowned for its high detection accuracy and computational efficiency, making it an ideal choice for isolating facial regions from cluttered backgrounds and standardizing the input for the classification model [6]. Second, for the core classification task, a Convolutional Neural Network (CNN) is utilized, specifically the *InceptionV3* architecture, given its documented success in face shape classification problems [7], [8], [9].

A significant contribution of this research is the introduction of a pseudo-labeling phase to enhance the quality of the training data. An initial model is trained and used to re-label the dataset to clean noisy labels and improve the overall robustness of the final classifier [10], [11]. Furthermore, to elevate model performance, the complex challenge of hyperparameter tuning is addressed. As manual tuning is tedious, a *Genetic Algorithm* (GA) is implemented to automate this optimization, navigating vast hyperparameter spaces to identify configurations that yield superior performance [12], [13], [14].

2. RESEARCH METHODS

The proposed hybrid intelligence framework is systematically constructed through a multi-stage process designed to ensure high accuracy and robustness in face shape classification. This section details the overall system architecture and the sequential stages of the research.

2.1. System Architecture Overview

The architecture of our proposed framework is conceptualized as a multi-stage, sequential pipeline that synergistically integrates robust preprocessing, iterative data refinement, and evolutionary optimization. The workflow begins with an input image, which is first processed for face detection using the Haar Cascade Classifier. If a face is successfully detected, the isolated facial region is passed to the next stage. The detected face is then classified by our final, optimized InceptionV3 model [8], [9], which has been rigorously trained on a refined dataset and tuned using a Genetic Algorithm to ensure high accuracy [7], [10]. The output of this classification determines the user's face shape, which serves as the basis for generating personalized eyewear recommendations.

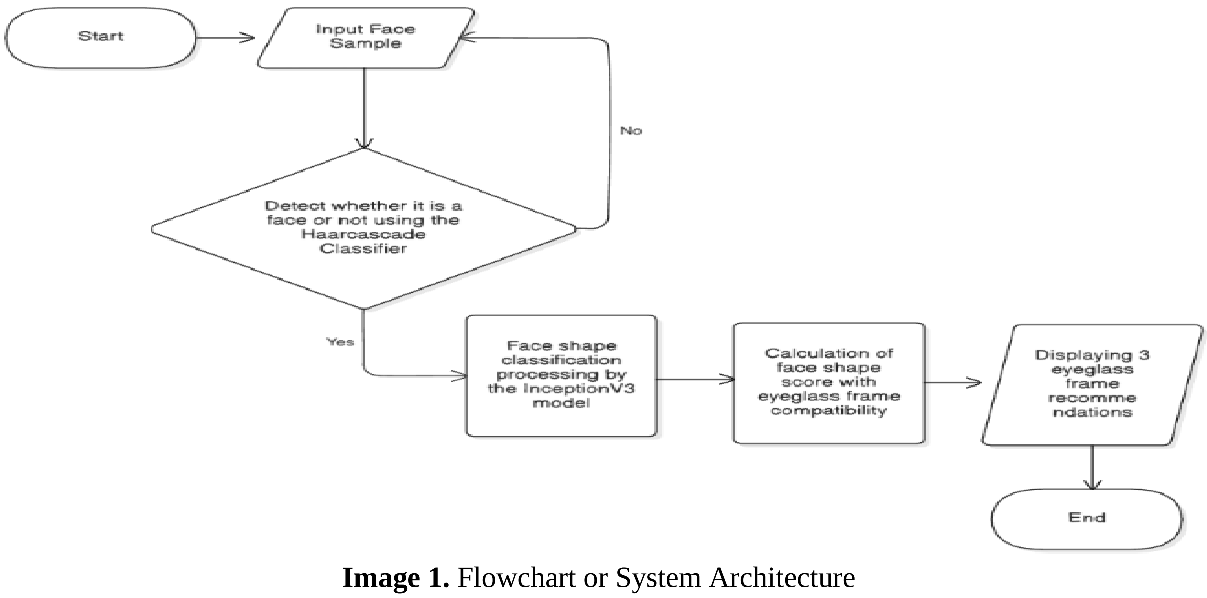

Image 1. Flowchart or System Architecture

2.2. Research Stages

The execution of this research was conducted through a sequence of distinct stages, ensuring a rigorous and reproducible workflow from data acquisition to final evaluation.

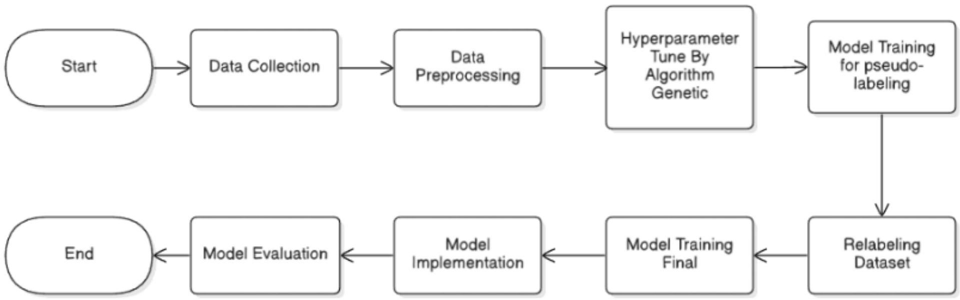

Image 2. Flowchart of the Research Stages

2.2.1. Data Collection

The initial phase of this research involved the curation of a robust and diverse dataset. This was achieved by amalgamating images from two extensive, publicly accessible repositories: Kaggle [15] and Roboflow [16]. The aggregation of these datasets was a deliberate strategy to create a more comprehensive corpus, featuring a wide spectrum of facial geometries, lighting conditions, ethnicities, and image qualities. The inherent heterogeneity of this combined dataset is fundamental to mitigating sampling bias and enhancing the generalization capabilities of the subsequently trained deep learning model [4], [2]. A diverse dataset compels the model to learn more invariant and robust features, which is critical for real-world application performance.

2.2.2. Data Preprocessing

After data collection, a meticulous data preprocessing pipeline was implemented to standardize the images and prepare them for model ingestion. This pipeline consisted of two critical steps:

a. Face Localization: The Viola-Jones algorithm, specifically its implementation using Haar-like features, known as the Haar Cascade Classifier, was employed for rapid and efficient facial detection [5]. This method is well-regarded for its high detection rates and computational efficiency, making it suitable for preprocessing large datasets [17]. By isolating and cropping the facial region from the original image, extraneous background information is eliminated, which reduces noise and allows the model to focus exclusively on relevant facial features [18].
b. Data Augmentation: To significantly increase the effective size of the training set and bolster the model's resilience to overfitting, a suite of data augmentation techniques was applied. These techniques included random geometric transformations such as rotation, scaling, and horizontal flipping, as well as photometric adjustments like changes in brightness and contrast. This process synthetically generates novel training instances, exposing the model to a wider range of variations and improving its ability to generalize from the training data to unseen images [19], [20].

2.2.3. Hyperparameter Tune by Algorithm Genetic

The performance of a Convolutional Neural Network (CNN) is highly dependent on the selection of its hyperparameters [21], [22]. Manual tuning of these parameters is often a laborious and suboptimal process [23]. Therefore, this study utilized a Genetic Algorithm (GA), a powerful metaheuristic optimization technique, to systematically and automatically explore the vast hyperparameter search space [24]. Key hyperparameters including learning rate, batch size, dropout probability, and the number and size of convolutional filters were encoded as a "chromosome" [25]. GA initialized a population of these chromosomes and iteratively evolved them over numerous generations using operations such as selection, crossover, and mutation [26]. The fitness of each chromosome was evaluated based on the validation accuracy of a CNN trained with the corresponding hyperparameters [7]. This evolutionary optimization process, conducted over a significant computational period, is highly effective at navigating complex, non-convex search spaces to identify a near-optimal hyperparameter configuration [27].

2.2.4. Model Training for pseudo-labeling

Leveraging the optimal hyperparameters identified by the GA, an initial "teacher" model was trained. This model was based on the InceptionV3 architecture, a deep CNN renowned for its high accuracy and computational efficiency in image classification tasks [1]. The choice of InceptionV3 was further supported by its successful application in various face-related computer vision problems [28]. This teacher model was trained on the preprocessed and augmented dataset until it converged. The primary objective of this model was not to serve as the final solution, but to act as a reliable, automated data labeler for the subsequent refinement stage [11].

2.2.5. Relabeling Dataset

To further enhance the quality of the training data, a pseudo-labeling strategy was implemented [10]. A standalone system was developed to apply the trained teacher model from the previous stage to the entire unlabeled training set. The model's predictions (the "pseudo-labels") were then used to replace the original ground-truth labels. This technique is particularly effective for correcting potential mislabels and reducing noise inherent in large, crowdsourced datasets [11]. By refining the dataset with high-confidence predictions from a capable teacher model, a cleaner and more consistent set of training data is generated, which is crucial for achieving higher performance in the final model [10].

2.2.6. Model Training Final

Using the refined, pseudo-labeled dataset, the final "student" model was trained. It is critical to note that this student model employed the identical InceptionV3 architecture and the same set of GA-optimized hyperparameters as the teacher model. This controlled approach ensures that any observed improvement in performance can be directly attributed to the enhanced quality of the training data rather than architectural or hyperparameter changes. The training process continued until the model's performance on a validation set plateaued, a significant improvement over the initial model.

2.2.7. Model Implementation

The fully trained and optimized model serves as the core of the end-to-end system for personalized eyewear recommendation. The system's workflow begins with the Haar Cascade module localizing the user's face, followed by the genetically tuned CNN classifying the face shape. To translate this classification into a practical recommendation, a knowledge-based engine utilizes a predefined scoring matrix. This matrix, derived from a user preference survey using google form, quantifies the suitability of various frame styles for each face shape.

Table 1. Frame Scoring Matrix

Frame Style	Oval	Heart	Oblong	Round	Square
Aviator	0.9	0.8	0.5	0.3	0.2
Rectangle	0.7	0.4	0.6	0.9	0.5
Cat Eyes	0.8	0.9	0.6	0.4	0.7
Round	0.6	0.5	0.7	0.2	0.9
Wayfarer	0.6	0.5	0.3	0.8	0.6
Square	0.6	0.3	0.4	0.9	0.1

Upon classification of a user's face shape, the system queries this matrix and recommends the frame style with the highest compatibility score. This final step transforms the model's abstract classification into a tangible, personalized, and actionable eyewear recommendation [2].

2.2.8. Model Evaluation

The final stage of the research was dedicated to a rigorous and unbiased evaluation of the final model's performance. A held-out test set, which was completely isolated and remained unseen during all preceding stages of training, optimization, and relabeling, was used for this purpose. This practice is essential to accurately assess the model's ability to generalize to new, previously unencountered data [14].

To provide a comprehensive assessment of the classifier's predictive capability, we employed four standard metrics: Accuracy, Precision, Recall, and F1-Score. These metrics provide a more nuanced view of performance than accuracy alone, particularly regarding class-specific correctness. They are mathematically defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

3. RESULT AND DISCUSSION

The Results and Discussion section presents the experimental findings and provides a comprehensive analysis of the proposed framework's performance. The discussion evaluates the effectiveness of the data preprocessing, the impact of the Genetic Algorithm (GA) on hyperparameter optimization, and the final classification accuracy achieved through pseudo-labeling.

3.1. Experiment Setup

The model was developed in Python using the TensorFlow library and executed on a high-performance computing environment equipped with an AMD Ryzen 5 4600H processor, 16 GB of RAM, and an NVIDIA GeForce GTX 1650 Ti GPU. The dataset was partitioned using an 80-10-10 split for training, validation, and testing, respectively [29].

3.2. Data Preprocessing

The foundational stage of our proposed framework involves rigorous data preprocessing to ensure the quality and consistency of the input data supplied to the Convolutional Neural Network (CNN). The primary objective of this stage is to perform facial localization, the process of accurately detecting and isolating the facial region from the surrounding background and non-essential elements within each image. For this critical task, we employed the Haar Cascade classifier, a robust and computationally efficient object detection method first proposed by Viola and Jones [30]. This method utilizes a cascade of classifiers trained with the Adaboost algorithm on a multitude of positive (images with faces) and negative (images without faces) samples, allowing for the rapid and accurate detection of frontal faces [31]. By isolating the facial region, we eliminate potential noise and variability introduced by background clutter, standardizing the input and ensuring that the subsequent CNN model focuses exclusively on the pertinent facial features necessary for shape classification

Image 3. Before Crop

Image 4. After Crop

The successful automation of this preprocessing step via the Haar Cascade classifier resulted in a refined collection of 14.925 cropped facial images. This final, standardized dataset, which includes undersampling across each class to ensure balance, serves as the direct input for the feature extraction and classification stages, providing a consistent and optimized foundation for training the genetically tuned CNN model.

3.3. Genetic Algorithm Hyperparameter Tuning

To optimize the performance of the InceptionV3 model, the research moved beyond manual tuning and employed a Genetic Algorithm (GA). This approach is highly effective for navigating the complex, high-dimensional hyperparameter search space characteristic of deep neural networks, a task where traditional methods often prove inefficient [7], [12]. The GA was tasked with systematically finding the optimal values for three critical hyperparameters: the learning rate, the dropout rate, and the number of neurons in the final dense layer, with the explicit goal of maximizing the model's validation accuracy.

The GA was implemented utilizing the DEAP framework, configured with a population size of 10 individuals evolving over 10 generations. Each individual in the population represented a unique set of CNN hyperparameters, specifically encoding three genes: the learning rate (ranging from $1x10^{-5}$ to $1x10^{-3}$), the dropout rate (from 0.2 to 0.7), and the number of neurons in the dense layer (from 128 to 512) [12], [25]. The evolutionary process was driven by standard genetic operators selected to effectively explore the hyperparameter space. A tournament selection strategy with a tournament size of 3 was employed to choose individuals for the next generation. For recombination, a Blend Crossover operator was used with a probability of 0.5 and an alpha value of 0.5. To introduce diversity, a Gaussian mutation operator was applied with an individual gene mutation probability of 0.2 [7], [26].

The fitness of each individual, and thus each hyperparameter set, was quantified by the validation accuracy achieved by the CNN model trained with its specific configuration [7]. The efficacy of this evolutionary process is demonstrated in Figure 5. Over the course of 10 generations, a clear upward trend in both the maximum and average validation accuracy was observed. This convergence pattern indicates that the GA successfully exploited promising regions of the search space, progressively evolving toward more optimal hyperparameter configurations.

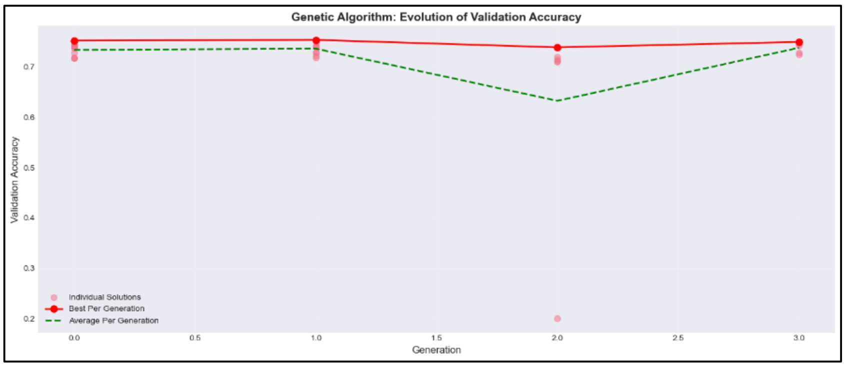

Image 5. Genetic Algorithm: Evolutionary validation Accuracy

The analysis of hyperparameter sensitivity, visualized in the correlation matrix (Image 6), revealed a moderate negative correlation (-0.29) between the learning rate and validation accuracy. This suggests that lower learning rates generally yielded better stability and performance within the explored range. Conversely, the dropout rate showed negligible correlation, highlighting the non-linear complexity of the tuning task [22].

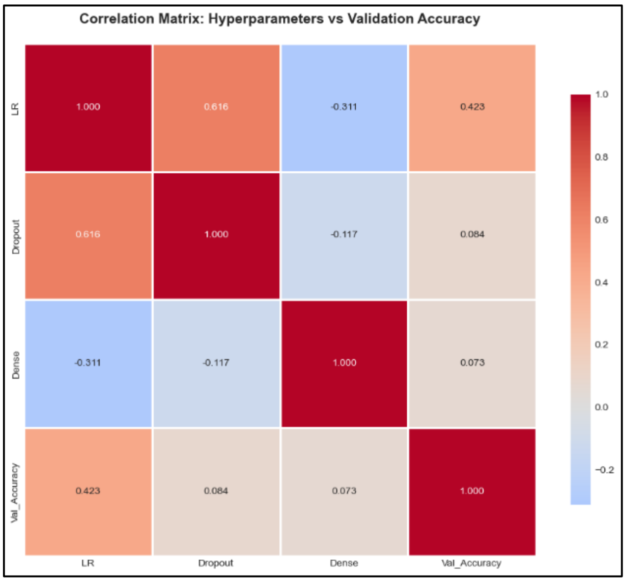

Image 6. Correlation Matrix: Hyperparameters vs Validation Accuracy

After the final generation, the GA identified the optimal set of hyperparameters that produced the highest validation accuracy of 75.27% at individual 18, generation 1. These values are detailed in Table 2

Table 2. Optimal Hyperparameters Identified by the Genetic Algorithm

Individual	Generation	Learning Rate	Dropout	Dense Unit	Best Validation Accuracy
1	0	0.00043	0.476	262	0.7176
2	0	0.000576	0.3916	271	0.7392
3	0	0.000898	0.6566	419	0.7419
4	0	0.000372	0.6465	273	0.7432
5	0	0.000909	0.692	376	0.7189
6	0	0.000543	0.3801	314	0.7284
7	0	0.000365	0.3692	190	0.7446
8	0	0.000858	0.3504	293	0.7378
9	0	0.000793	0.4035	243	0.7176
10	0	0.000124	0.3178	487	0.7527
11	1	0.000205	0.2995	433	0.7284
12	1	0.000284	0.3875	244	0.723
13	1	0.000124	0.3178	487	0.7405
14	1	0.000124	0.3178	487	0.7189
15	1	0.000365	0.3692	190	0.7514
16	1	0.000365	0.3692	190	0.7432
17	1	0.000124	0.2606	487	0.7297
18	1	0.000297	0.347	316	0.7541
19	1	0.000272	0.3217	306	0.7405
20	2	-0.000223	0.3318	246	0.2
21	2	0.000095	0.3383	542	0.7122

22	2	0.000057	0.3509	486	0.7108
23	2	0.000432	0.3362	191	0.7392
24	2	0.000124	0.3178	487	0.7203
25	2	0.000124	0.3178	487	0.7149
26	3	0.00039	0.3634	147	0.75
27	3	0.000272	0.3529	359	0.7284
28	3	0.00028	0.3477	129	0.7243
29	3	0.000449	0.3355	377	0.7432
30	3	0.000422	0.3292	190	0.7459

In conclusion, the systematic and automated approach of the Genetic Algorithm proved instrumental in identifying a highly effective set of hyperparameters. This optimized configuration forms the basis for our final model, ensuring it is robustly tuned for the face shape classification task.

3.4. Development of the Teacher Model for Pseudo-Labeling

Following the hyperparameter optimization by the Genetic Algorithm, an initial InceptionV3 model was trained. This model's designated role was to serve as a "teacher" in our pseudo-labeling framework [10]. The objective was to create a competent, though not final, model capable of re-annotating the training dataset with high-confidence predictions, thereby refining the ground-truth labels for the subsequent training of a final "student" model. This teacher model was trained for 100 epochs using the GA-tuned hyperparameters. The training and validation accuracy and loss curves, depicted in Image 7, show a consistent learning trajectory, culminating in a final test accuracy of 88.54%.

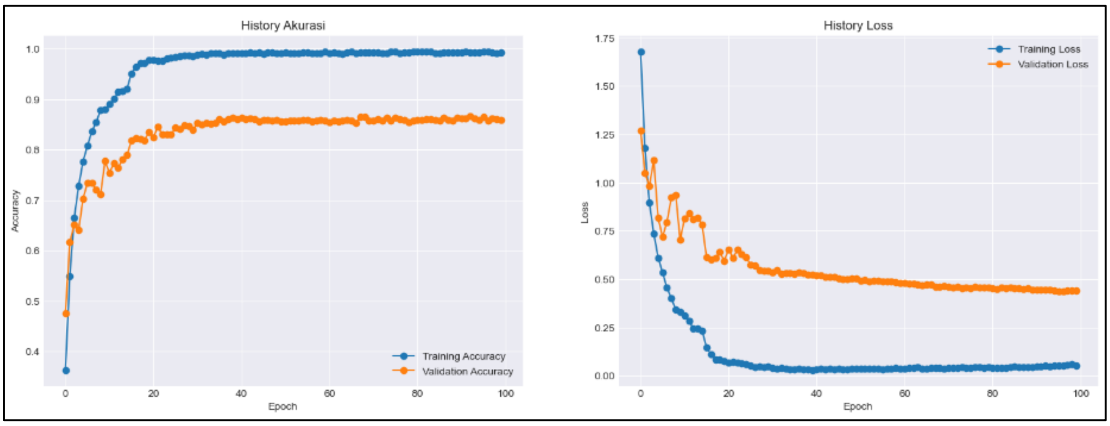

Image 7. Training and Validation Curves of the Teacher Model

A granular analysis of the teacher model's performance is presented in the confusion matrix Image 8 and the accompanying classification report. The matrix reveals strong predictive capability across all five face shape classes, with most predictions falling correctly on the diagonal. The classification report further quantifies this, showing a weighted average precision and recall of 0.89. While highly effective, the model exhibits minor confusion between certain classes, such as its lower recall (0.81) on "Oblong" faces. This specific performance profile makes it an ideal candidate for pseudo-labeling; it is accurate enough to generate reliable labels for the majority of the dataset, effectively cleaning and enhancing the data that the final student model will learn from [11], [32].

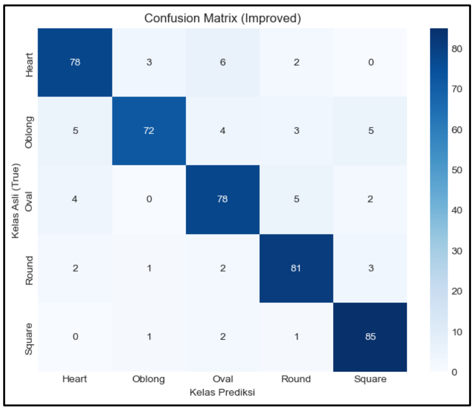

Image 8. Confusion Matrix of the Teacher Model

	precision	recall	f1-score	support
Heart	0.88	0.88	0.88	89
Oblong	0.94	0.81	0.87	89
Oval	0.85	0.88	0.86	89
Round	0.88	0.91	0.90	89
Square	0.89	0.96	0.92	89
accuracy			0.89	445
macro avg	0.89	0.89	0.88	445
weighted avg	0.89	0.89	0.88	445

Image 9. Classification Report of the Teacher Model

3.5. Relabeling Dataset

Upon developing an initial baseline model with a commendable test accuracy of 88.54%, a strategic data refinement phase was initiated using a pseudo-labeling methodology [10]. This technique leverages the trained model, referred to as the "teacher model," to re-evaluate and cleanse the entire dataset, including the training, validation, and testing subsets. The core principle of this process is to enhance the quality and consistency of the data labels, thereby mitigating the impact of potential noise and labeling inaccuracies inherent in the original dataset [11]. The procedure involved iterating through every image in the dataset and subjecting it to prediction by the teacher model. If the predicted label for an image did not correspond with its assigned ground-truth label, the image was deemed ambiguous or mislabeled. Consequently, such conflicting instances were systematically eliminated from the dataset. This data pruning strategy ensures that the subsequent model is trained on a higher-fidelity dataset, which is hypothesized to lead to a more robust and accurate final classifier. The reduction in dataset size per class following this filtering process is visualized in Image 10.

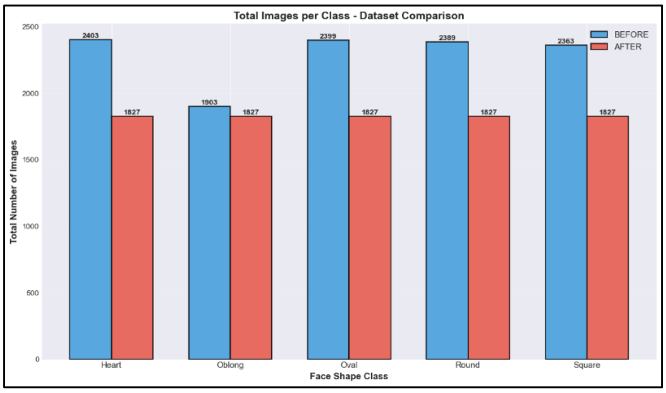

Image 10. Total Images Per Class Comparison Before and After Relabeling

The quantitative impact of the relabeling process was substantial, resulting in a more streamlined and perfectly balanced dataset. Initially, the dataset comprised 11,457 images with a noticeable class
imbalance across the data splits. After the teacher model-based filtering, the total number of images was reduced to 9,135, representing a focused dataset pruned of 2,322 ambiguous samples. The most significant outcome of this procedure was the achievement of perfect class balance. In the refined dataset, each of the five face shape classes (Heart, Oblong, Oval, Round, and Square) constitutes precisely 20% of the images
within the training, validation, and testing sets. This transformation from an imbalanced to a perfectly balanced distribution, as illustrated in Image 11, is critical for preventing model bias and ensuring that the classifier learns to distinguish each class with equal proficiency. This refined and

balanced dataset serves as the definitive foundation for the subsequent hyperparameter optimization and final model training stages.

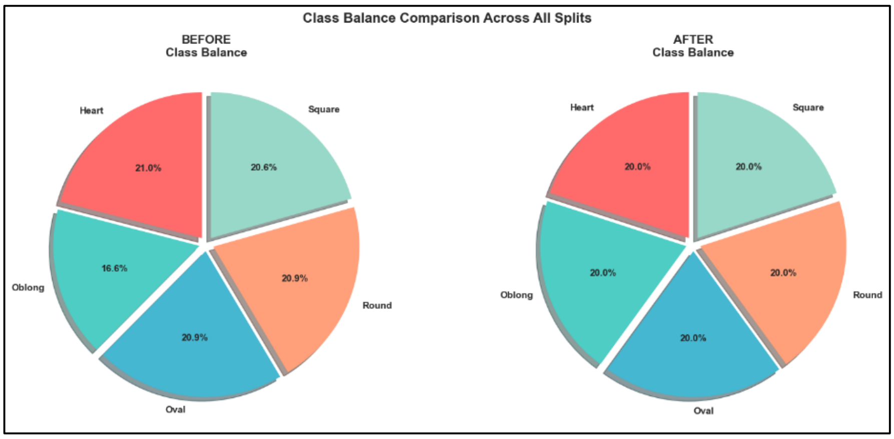

Image 11. Class Balance Comparison Before and After Relabeling

3.6. Final Model Training and Evaluation

Following the dataset refinement via pseudo-labeling, the final "student" model was trained. This model employed the identical InceptionV3 architecture and the GA-optimized hyperparameters used by the teacher model to ensure that performance gains were directly attributable to data quality rather than architectural changes.

The training commenced with the refined dataset produced through pseudo-labeling. An initial baseline model, which achieved a test accuracy of 88.54%, was first employed to re-label the training dataset. This technique, known as pseudo-labeling, serves to correct potential labeling errors and enhance data consistency, thereby providing a cleaner and more reliable dataset for the final model [10], [11]. The final model was subsequently trained on this purified dataset, a critical step for achieving superior performance.

The model was scheduled to train for 100 epochs, but training was concluded at 33 epochs due to the implementation of an early stopping criterion, with its performance monitored on both the training and validation datasets. As shown in image 12, the accuracy curves for both training and validation exhibit a steady learning progression, while the loss curves show a consistent decline, which signifies a successful convergence without substantial overfitting.

Image 12. Training Log in Early Epoch

The learning rate was dynamically adjusted during training, as seen in the training logs, beginning at $2.97 \, x \, 10^{-4}$ and being reduced at subsequent plateaus to fine-tune the model's weights, a common practice to enhance model performance [33].

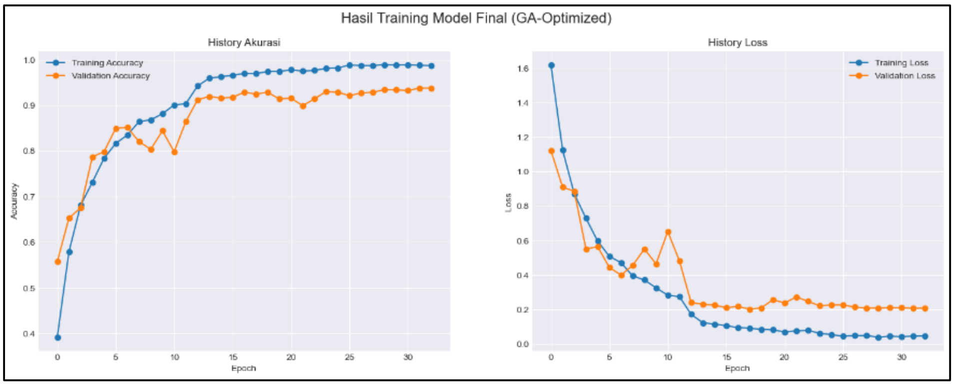

Image 13. Training and Validation Curves of the Final Model

Upon completion of the training, the model was subjected to a final evaluation on a held-out test set. The model achieved a final test accuracy of 97.65%, a substantial improvement of 9.11 percentage points over the initial baseline. This result strongly validates the efficacy of the proposed hybrid approach, where GA-based optimization and pseudo-labeling synergistically enhance classification performance.

The detailed performance across the five face shape classes (Heart, Oblong, Oval, Round, Square) is presented in the confusion matrix in Image 14 and the classification report in Image 15. The confusion matrix displays a strong diagonal, indicating a high rate of correct classifications for all classes with minimal confusion between them. The classification report further corroborates with this, showing excellent precision, recall, and F1-scores, all approximately 0.98 on a weighted average basis. This balanced, high-level performance across all classes underscores the model's robustness and its suitability for deployment in a real-world personalized eyewear recommendation system [3], [8].

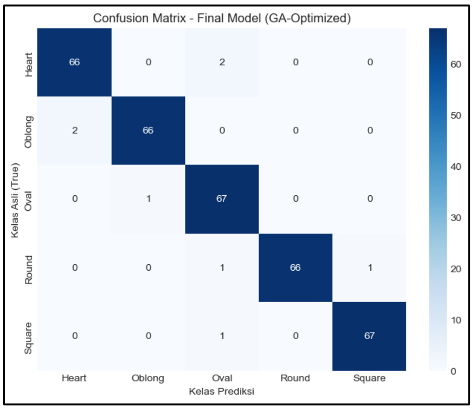

Image 14. Confusion Matrix of the Final Model

	precision	recall	f1-score	support
Heart	0.97	0.97	0.97	68
Oblong	0.99	0.97	0.98	68
Oval	0.94	0.99	0.96	68
Round	1.00	0.97	0.99	68
Square	0.99	0.99	0.99	68
accuracy			0.98	340
macro avg	0.98	0.98	0.98	340
weighted avg	0.98	0.98	0.98	340

Image 15. Classification Report of the Final Model

3.7. Model Comparison

To contextualize our framework's performance, we benchmarked our initial and genetically optimized models against established face classification methods from the literature. Table 3 summarizes this comparative analysis, highlighting the efficacy of our approach against other deep learning and evolutionary computation techniques.

Table 3. Model Comparison

Model	Best Accuracy
GA-tuned CNN [7]	94.50%
Retrained InceptionV3 [8]	92.65%
Our Final Model	97.65%
Our Initial Model	88.54%

3.8. System Testing

The system testing phase evaluates the real-time performance and user interaction flow of the integrated framework. Upon activation, the system initializes the camera feed and begins real-time face localization using a dual-validation approach with Haar Cascade classifiers, which require simultaneous detection of a frontal face and at least one eye to confirm the presence of a valid subject [34]. A graphical user interface (GUI) provides a guide, prompting the user to center their face within a predefined boundary; the capture functionality remains disabled until this condition is met to ensure standardized input. Once the user initiates the capture, the system automatically acquires three distinct image samples in quick succession.

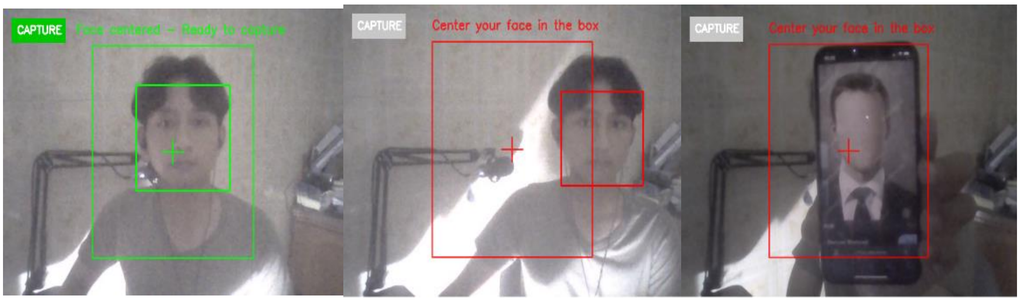

Image 16. Testing System with Realtime Face Detection

Each sample is independently fed into the genetically tuned CNN classifier to generate a prediction vector of face shape probabilities. To enhance robustness and minimize single-frame anomalies, these three vectors are averaged to produce a final, stabilized probability distribution. This average result is then algorithmically matched against a predefined JSON profile containing compatibility scores for various eyewear styles. Based on a weighted sum calculation, the system identifies and displays the top three most suitable eyewear frame recommendations, ranked by their compatibility score, thus completing the personalized recommendation process.

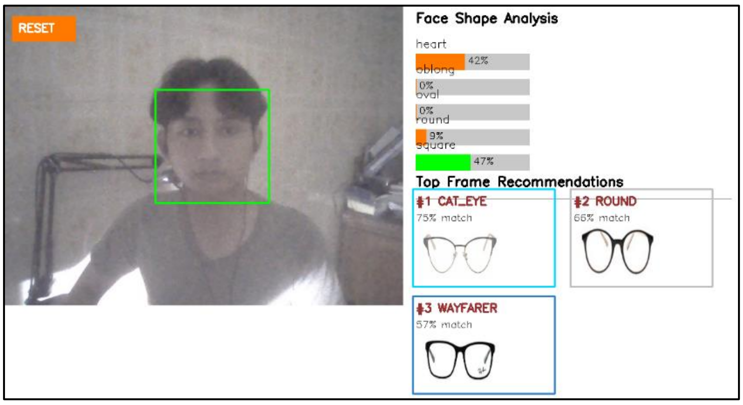

Image 17. Testing System Result Recommendations

4. CONCLUSION

This study successfully developed a hybrid intelligence framework for personalized eyewear recommendation by integrating a Haar Cascade classifier for face localization, a pseudo-labeling strategy for data refinement, and a Convolutional Neural Network (CNN) based on the InceptionV3 architecture for classification. The application of a Genetic Algorithm (GA) to systematically optimize the CNN's hyperparameters proved critical, elevating the model's test accuracy from an initial 88.54% to a final 97.65%. This significant improvement demonstrates the powerful synergy between deep learning and evolutionary computation, establishing a new benchmark for accuracy in personalized recommendation systems.

For further development, future research should focus on optimizing the proposed model for deployment on resource-constrained edge devices, such as mobile phones, by exploring lightweight architectures or model quantization techniques. Additionally, while the current dataset is robust, expanding it to include a broader diversity of facial ethnicities and extreme lighting conditions would further enhance the system's generalization capabilities and robustness in diverse real-world scenarios.

References

[1] V. Ramadhan, M. W. Sardjono, M. Cahyanti, R. Swedia, M. Ridwan, and D. Septian, “Klasifikasi Bentuk Bingkai (Frame) Kacamata Menggunakan CNN dengan Arsitektur Inception V3 dan Augmented Reality Berbasis Android,” *Journal of System and Computer Engineering (JSCE) ISSN*, vol. 5, no. 2, p. 204, Apr. 2024, doi: https://doi.org/10.61628/jsce.v5i2.1292.

[2] J. Zhao, M. Zhang, C. He, X. Xie, and J. Li, “A novel facial attractiveness evaluation system based on face shape, facial structure features and skin,” *Cogn Neurodyn*, vol. 14, no. 5, pp. 643–656, Oct. 2020, doi: 10.1007/s11571-020-09591-9.

[3] A. Sukumaran and T. Brindha, “Optimal feature selection with hybrid classification for automatic face shape classification using fitness sorted Grey wolf update,” *Multimed Tools Appl*, vol. 80, no. 17, pp. 25689–25710, Jul. 2021, doi: 10.1007/s11042-021-10710-9.

[4] J. Khatib Sulaiman Dalam No, N. Dakhil Hasan, and A. M. Abdulazeez, “Face Recognition Based on Deep Learning: A Comprehensive Review,” *Indonesian Journal of Computer Science*, vol. 13, no. 3, Jun. 2024, doi: https://doi.org/10.33022/ijcs.v13i3.4037.

[5] F. T. Nugroho and E. I. Sela, “Deteksi Citra Wajah Menggunakan Algoritma Haar Cascade Classifier,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 1, pp. 37–44, Dec. 2023, doi: 10.57152/malcom.v4i1.988.

[6] E. Setyaningsih and L. Widodo, “Comparative Analysis of Haar Cascade Classifier, Dlib, and Mediapipe for Face Recognition Analisis Perbandingan Haar Cascade Classifier, Dlib, dan Mediapipe untuk Pengenalan Wajah,” *ELECTRON Jurnal Ilmiah Teknik Elektro*, vol. 6, no. 1, May 2025, doi: https://doi.org/10.33019/electron.v6i1.240.

[7] N. Karlupia, P. Mahajan, P. Abrol, and P. K. Lehana, “A Genetic Algorithm Based Optimized Convolutional Neural Network for Face Recognition,” *International Journal of Applied Mathematics and Computer Science*, vol. 33, no. 1, pp. 21–31, Mar. 2023, doi: 10.34768/amcs-2023-0002.

[8] T. Hidayat, I. A. Astuti, A. Yaqin, A. Phuk Tjilen, and T. Arifianto, “Grouping of Image Patterns Using Inceptionv3 For Face Shape Classification,” *Journal on Informatics Visualization*, vol. 7, Dec. 2023, doi: http://dx.doi.org/10.62527/joiv.7.4.1743.

[9] A. Emmanuel, C. Tio, and D. C. Tio, “Face shape classification using Inception v3,” *ArXiv*, Nov. 2019, doi: 10.48550/arXiv.1911.07916.

[10] P. Kage, J. C. Rothenberger, P. Andreadis, and D. I. Diochnos, “A Review of Pseudo-Labeling for Computer Vision,” *ArXiv*, May 2025, [Online]. Available: http://arxiv.org/abs/2408.07221

[11] H. Pham, Z. Dai, Q. Xie, M.-T. Luong, and Q. V. Le, “Meta Pseudo Labels,” *ArXiv*, Mar. 2021, [Online]. Available: http://arxiv.org/abs/2003.10580

[12] W. Hussain *et al.*, “Ensemble genetic and CNN model-based image classification by enhancing hyperparameter tuning,” *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-024-76178-3.

[13] P. Kalaiarasi and P. E. Rani, “A Novel Approach for Optimization of Convolution Neural Network with Particle Swarm Optimization and Genetic Algorithm for Face Recognition,”

International Journal on Recent and Innovation Trends in Computing and Communication, vol. 11, no. 4, pp. 215–223, Mar. 2023, doi: 10.17762/ijritcc.v11i4s.6531.
[14] P. Ghosh, S. Jana, S. Ghosh, and R. Dhar, "Hyperparameter Optimization in Deep Learning Techniques for Multimodal Biometric Verification," *Devotion Journal of Research Progress*, vol. 6, no. 9, 2025, doi: https://doi.org/10.59188/devotion.v6i9.25529.
[15] Niten Lama, "Face Shape Dataset," *Kaggle*, Jan. 2020.
[16] Project, "Face Shape Classification Dataset," *Roboflow Universe*, Sep. 2023.
[17] W. C. Cheng, H. C. Hsiao, Y. Q. Hong, and D. Y. Wang, "Masked face recognition based on facenet and genetic algorithm," *International Journal of Applied Science and Engineering*, vol. 20, no. 3, 2023, doi: 10.6703/IJASE.202309_20(3).005.
[18] I. T. Aruleba and Y. Sun, "Deep Learning and Genetic Algorithms Approach for Age Estimation Based on Facial Images," *International Journal of Computer Theory and Engineering*, vol. 16, no. 4, pp. 127–133, 2024, doi: 10.7763/IJCTE.2024.V16.1360.
[19] F. Fatimatuzzahra, L. Lindawati, and S. Soim, "Development of Convolutional Neural Network Models to Improve Facial Expression Recognition Accuracy," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 10, no. 2, pp. 279–289, Jun. 2024, doi: 10.26555/jiteki.v10i2.28863.
[20] O. Oladipo, E. O. Omidiora, and V. C. Osamor, "A novel genetic-artificial neural network based age estimation system," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-23242-5.
[21] P. Probst and B. Bischl, "Tunability: Importance of Hyperparameters of Machine Learning Algorithms," *Journal of Machine Learning Research*, vol. 20, pp. 1–32, 2019, doi: https://doi.org/10.48550/arXiv.1802.09596Focustolearnmore.
[22] J. A Ilemobayo *et al.*, "Hyperparameter Tuning in Machine Learning: A Comprehensive Review," *Journal of Engineering Research and Reports*, vol. 26, no. 6, pp. 388–395, Jun. 2024, doi: 10.9734/jerr/2024/v26i61188.
[23] D. Passos and P. Mishra, "A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks," Apr. 15, 2022, *Elsevier B.V.* doi: 10.1016/j.chemolab.2022.104520.
[24] X. Wang and Y. Li, "Facial Recognition System Based on Genetic Algorithm Improved ROI-KNN Convolutional Neural Network," *Appl Bionics Biomech*, vol. 2022, Oct. 2022, doi: 10.1155/2022/7976856.
[25] M. Sam'an, Safuan, and M. Munsarif, "Convolutional neural network hyperparameters for face emotion recognition using genetic algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 442–449, Jan. 2024, doi: 10.11591/ijeecs.v33.i1.pp442-449.
[26] V. Kachitvichyanukul, "Comparison of Three Evolutionary Algorithms: GA, PSO, and DE," *Industrial Engineering and Management Systems*, vol. 11, no. 3, pp. 215–223, Sep. 2012, doi: 10.7232/iems.2012.11.3.215.
[27] M. Munsarif, E. Noersasongko, P. N. Andono, and M. A. Soeleman, "Improving convolutional neural network based on hyperparameter optimization using variable length genetic algorithm for english digit handwritten recognition," *International Journal of Advances in Intelligent Informatics*, vol. 9, no. 1, pp. 66–78, Mar. 2023, doi: 10.26555/ijain.v9i1.881.
[28] S. Degaonkar and A. Agarkar, "Real Time Face Mask Detection Using Inception-V3," *Int J Res Appl Sci Eng Technol*, vol. 11, no. 11, pp. 973–979, Nov. 2023, doi: 10.22214/ijraset.2023.56652.
[29] G. Elvis, "Optimization of CNN Pipelines for Low-light Image Recognition Using Genetic Algorithms," *SSRN Electronic Journal*, Sep. 2020, doi: 10.2139/ssrn.5289444.
[30] M. S. Nidom, "Haar Cascade Classifier and Adaboost Algorithm for Face Detection with the Viola-Jones Method," *Transactions on Informatics and Data Science*, vol. 2, no. 1, pp. 15–26, Mar. 2025, doi: 10.24090/tids.v2i1.12276.
[31] D. R. Rochmawati, "Deteksi Wajah Dengan Metode Haar Cascade Menggunakan OpenCV (Face Detection With Haar Cascade Method Using OpenCV)," *Journal Politeknik Pajajaran*, Sep. 2024, doi: 10.59820/tekomin.v3i1.293.

[32] H. M. Al-Dabbas, R. A. Azeez, and A. E. Ali, "Two Proposed Models for Face Recognition: Achieving High Accuracy and Speed with Artificial Intelligence," *Engineering, Technology and Applied Science Research*, vol. 14, no. 2, pp. 13706–13713, Apr. 2024, doi: 10.48084/etasr.7002.
[33] H. Amri, "Real-Time Face Attendance System Using CNN Mobilenet and MTCNN," *JURNAL TEKNOLOGI DAN OPEN SOURCE*, vol. 8, no. 1, pp. 164–179, Jun. 2025, doi: 10.36378/jtos.v8i1.4403.
[34] N. A. Mohd Ariffin, U. A. Gimba, and A. Musa, "Face Detection based on Haar Cascade and Convolution Neural Network (CNN)," *Journal of Advanced Research in Computing and Applications*, vol. 38, no. 1, pp. 1–11, Mar. 2025, doi: 10.37934/arca.38.1.111.