P-ISSN: 2656-1743 | E-ISSN: 2656-1735 **JURNAL RISET INFORMATIKA**
DOI: https://doi.org/10.34288/jri.v8i1.465 Vol. 8, No. 1. December 2025
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

FACIAL RECOGNITION PERFORMANCE EVALUATION WITH YOLOV8, ARCFACE, AND SVM IN A CONTACTLESS EMPLOYEE ATTENDANCE SYSTEM

Glanes Cindy Terampe[-1], Arif Pramudwiatmoko[-2]

Program Studi Informatika
Universitas Teknologi Yogyakarta
glanesscindy@gmail.com[-1], arif.pramudwiatmoko@uty.ac.id[-2]

Abstract

Manual attendance systems, which continue to be implemented in many institutions, are vulnerable to manipulation and require significant time. This research proposes an automated facial recognition attendance system optimized to address the unique challenges posed by CCTV cameras installed at a height of 3 meters. The system integrates three main components: YOLOv8m for face detection, ArcFace for 512-dimensional feature extraction, and a Support Vector Machine (SVM) with a Polynomial kernel for identity classification. The dataset (5 classes) was augmented using 20 augmentations per image and was split into a 70% training and 30% testing ratio. An image preprocessing pipeline, including CLAHE, denoising, and sharpening, was applied to enhance the input image quality. Experimental results demonstrate high classification performance, achieving 93.7% accuracy, 0.938 precision, 0.937 recall, and an F1-Score of 0.935. Confusion matrix and PCA analysis identified that the primary misclassification occurred between the E005_employee5 and E002_employee2 classes, correlating with feature overlap. Computationally, the system achieved a throughput of 7.2 FPS on the testing hardware. The system is proven to be accurate and functional for the attendance task, although its real-time performance (FPS) is highly dependent on hardware acceleration.

Keywords: Deep Learning; Face Recognition; YOLOv8; ArcFace; Support Vector Machine

Abstrak

Sistem presensi manual yang masih diterapkan di banyak institusi rentan terhadap manipulasi dan memerlukan waktu yang signifikan. Penelitian ini mengusulkan sistem presensi otomatis berbasis pengenalan wajah yang dioptimalkan untuk mengatasi tantangan unik dari kamera CCTV yang dipasang pada ketinggian 3 meter. Sistem ini mengintegrasikan tiga komponen utama: YOLOv8m untuk deteksi wajah, ArcFace untuk ekstraksi fitur 512-dimensi, dan Support Vector Machine (SVM) dengan kernel Polinomial untuk klasifikasi identitas. Dataset (5 kelas) diperbanyak menggunakan 20 augmentasi per gambar dan dibagi dengan rasio 70% data latih dan 30% data uji. Alur prapemrosesan citra, termasuk CLAHE, denoising, dan sharpening, diterapkan untuk meningkatkan kualitas gambar input. Hasil eksperimen menunjukkan kinerja klasifikasi yang tinggi, mencapai akurasi 93.7%, presisi 0.938, recall 0.937, dan F1-Score 0.935. Analisis confusion matrix dan PCA berhasil mengidentifikasi misklasifikasi utama terjadi antara kelas E005_employee5 dan E002_employee2, yang berkorelasi dengan tumpang tindih fitur. Secara komputasi, sistem mencapai throughput 7.2 FPS pada perangkat keras pengujian. Sistem ini terbukti akurat dan fungsional untuk tugas presensi, meskipun performa real-time (FPS) sangat bergantung pada akselerasi perangkat keras.

Kata kunci: Deep Learning; Pengenalan Wajah; YOLOv8; ArcFace; Support Vector Machine

INTRODUCTION

Employee attendance management, or attendance, is a fundamental aspect of organizational administration crucial for monitoring discipline and productivity (Armiady, 2021), (Sim et al., 2024). In high-demand environments such as hospitals, accurate and

efficient attendance tracking becomes essential (Armiady, 2021; Romli, 2021). Conventional attendance systems, such as manual signatures, have significant drawbacks (Walangitan et al., 2024). These methods are not only vulnerable to data manipulation, such as the practice of "buddy punching" (Walangitan et al., 2024), (Adim & Nurhidayat, 2023; Nawawi, 2023), but are also time-consuming and complicate the data recapitulation process (Nawawi, 2023), (Arishadilah, 2024). Other biometric methods like fingerprints, although more modern, require physical contact. This poses hygiene risks, a concern brought to the forefront post-COVID-19 pandemic (Mahalwal et al., 2020), and is also prone to failure if the user's fingerprints are dirty or injured (Nawawi, 2023).

To address these limitations, automated attendance systems based on computer vision using facial recognition (FR) have emerged as a superior alternative solution (Dony & Lubis, 2025). This technology offers a non-contact attendance method that can automate the attendance tracking process in real-time (Dumbere et al., 2024), (Adelia et al., 2021). Many institutions, including hospitals, already possess security infrastructure in the form of surveillance cameras (Closed-Circuit Television or CCTV) (Tofir et al., 2020), (Listyoningrum et al., 2023). Utilizing "Smart CCTV" powered by Artificial Intelligence (AI) for facial recognition enables the existing surveillance system to not only function for security but also to automatically record attendance (Ren et al., 2025), (Tofir et al., 2020).

Nevertheless, the implementation of facial recognition in real-world environments such as hospitals presents complex technical challenges. Traditional FR systems are often static, requiring users to actively cooperate with the camera (Fan et al., 2021). Conversely, footage from surveillance CCTV operates in unconstrained conditions (Fan et al., 2021). This challenge is exacerbated by the common placement of CCTV cameras in public facilities, namely at a high position (estimated at 3 meters in this research) to gain wide surveillance coverage. This installation results in capturing low-resolution and small-sized faces. These factors, coupled with real-world conditions such as pose variations, inconsistent illumination, and occlusion (faces being obstructed by objects or mask usage), can significantly degrade detection and recognition accuracy (Yu & Zhang, 2021), (Alzu'bi et al., 2021), (Hasan & Hardjianto, 2024).

To address this challenge, this research proposes an integrated deep learning pipeline that combines three state-of-the-art methods. The first stage is face detection and segmentation, which uses the YOLOv8 (You Only Look Once version 8) algorithm. YOLO is known as an advanced real-time object detection algorithm due to its balance between speed and accuracy (Terven & Cordova-Esparza, 2023), (Zhang et al., 2023). YOLOv8, released in 2023, demonstrates significant accuracy (mAP) improvements over its predecessors, such as YOLOv5 (Yisihak & Li, 2024), (Karakuş et al., 2023), (Yuan et al., 2024). Its new anchor-free and decoupled head architecture, along with its segmentation capabilities, makes it ideal for detecting and isolating faces in complex scenarios (Nurlita et al., 2024).

Second, once the face is detected, facial features are extracted using ArcFace. ArcFace is a highly robust facial recognition model that utilizes Additive Angular Margin Loss. This approach is designed to maximize inter-class feature discrimination (different identities) and minimize intra-class variation (Alzu'bi et al., 2021), (Sydor et al., 2024). In comparative studies, ArcFace has been shown to provide superior performance (e.g., a True Positive Rate of 0.92) compared to other models such as FaceNet or VGGFace (Sydor et al., 2024), and demonstrates high accuracy even on limited datasets (Mustafa Abdullah & Mohsin Abdulazeez, 2021).

The final stage is identity classification. The feature embeddings from ArcFace are then classified using a Support Vector Machine (SVM). SVM is an efficient and reliable supervised classification method (Nguyen et al., 2023), (Marappan et al., 2021). SVM has proven successful and has achieved high accuracy in various machine learning applications, including facial recognition (Marappan et al., 2021), (Valero-Carreras et al., 2023).

This research aims to design and implement an automated, real-time, and accurate hospital employee attendance system by leveraging existing CCTV infrastructure installed at a height (±3 meters). The main contribution of this study is the design and evaluation of a hybrid architecture that integrates YOLOv8 (for detection and segmentation), ArcFace (for feature extraction), and SVM (for classification). This combination is specifically designed to overcome the difficult challenges in real-world surveillance scenarios, particularly the problems arising from high-mounted cameras, such as low facial resolution, extreme pose angles, and occlusion.

Operationally, the system is designed to function automatically and unobtrusively by leveraging existing CCTV cameras installed in strategic areas such as workroom entrances or

hospital corridors. Each time an employee enters the surveillance zone, the camera captures real-time video. Every video frame is then processed through the proposed pipeline: YOLOv8 detects faces from the high-angle view, ArcFace extracts facial features from the quality-enhanced images via the image preprocessing pipeline, and SVM classifies the employee's identity based on the generated embedding. If a face is successfully recognized and matches a predefined work schedule, the system automatically records the attendance in the database without requiring physical interaction or user awareness. If the face is not recognized or the employee is not on schedule, the system logs it as "unknown" or "no scheduled shift" for further review. With this approach, the system not only replaces conventional methods that are prone to manipulation but also offers a hygienic, efficient solution that can be seamlessly integrated with existing security infrastructure.

RESEARCH METHODS

This research methodology is designed to develop and evaluate an end-to-end facial recognition attendance system. The research stages encompass the entire system workflow, starting from raw data acquisition from CCTV to identity classification and attendance recording.

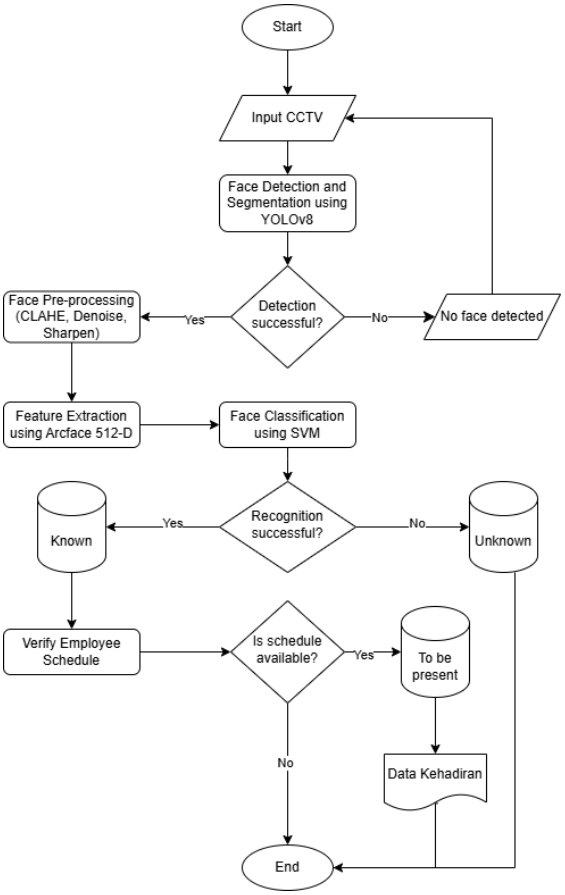


Figure 1. Flowchart of the Research Methodology for an Employee Attendance System Using CCTV with YOLOv8, ArcFace, and SVM

The proposed research methodology for developing the automated attendance system is comprehensively illustrated in Figure 1. The system workflow begins with the acquisition of video frames received in real-time from the CCTV input. Each frame is then processed by the YOLOv8 face detection model to localize and segment the facial region. If detection is successful, the cropped facial image undergoes an intensive pre-processing stage consisting of Contrast Limited Adaptive Histogram Equalization (CLAHE) for contrast enhancement, denoising (Median Blur) for noise reduction, and sharpening (2D Filter) for detail enhancement. Following image quality optimization, the system applies the ArcFace feature extraction model to transform the facial image into a unique 512-dimensional vector embedding. This feature vector is then fed into a pre-trained Support Vector Machine (SVM) classifier to perform the identification process. If the SVM successfully recognizes the face with a sufficient confidence level (Known), the system proceeds to the business

logic to validate the employee's schedule. If the schedule is confirmed, the attendance status (Attendance) is recorded in the Attendance Database. Conversely, if the face is not recognized (Unknown) or the employee does not have a schedule, the system records this data separately and completes the processing cycle.

Data Acquisition and Pre-processing

The primary data source is a real-time video feed from a CCTV camera mounted on the ceiling or upper wall at an estimated height of 3 meters from the floor. Acquisition from this high angle presents significant challenges, including low facial resolution, perspective distortion, and extreme pose angles (Alzu'bi et al., 2021). Examples of facial conditions captured from this height can be seen in Figure 2, which shows small variations in face size, a top-down viewing angle, and uneven lighting conditions.

Figure 2. Example of CCTV Frame from 3 Meters Height. The faces appear small with a top-down viewing angle and varying lighting conditions.

The training dataset was collected from employee video recordings, from which frames were extracted. Each frame then went through pre-processing using a consistent pipeline for training and inference. As implemented in the system, the pre-processing pipeline includes three main stages:

1) CLAHE Enhancement

Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to the luminance (Y) channel in the YCrCb color space to enhance local contrast while limiting noise amplification (Hasan & Hardjianto, 2024). This transformation is defined as:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \qquad (1)$$

Where:

- s_k is the output pixel intensity (the transformation result).
- r_k is the input (original) pixel intensity at level *k.*
- L is the total number of intensity levels (e.g., 256 for an 8-bit image).
- $p_r(r_j)$ is the probability of occurrence (the normalized histogram value) of pixels with intensity *j.*
- $\sum_{j=0}^{k} p_r(r_j)$ is the Cumulative Distribution Function (CDF) of the input pixel intensity, limited by clipLimit=2.0

2) Denoising

A Median filter with a 3×3 kernel is applied to remove salt-and-pepper noise while preserving edge information (Yisihak & Li, 2024).

3) Sharpening

A Laplacian sharpening kernel is applied to enhance facial details:

$$K_{sharpen} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad (2)$$

Data augmentation was used to enhance the model's robustness against variations in pose, illumination, and real-world conditions. Augmentations were applied using the *imgaug* library, featuring transformations such as: horizontal flip (probability 0.5), random rotation (between -15° and +15°), scaling (between 0.9× and 1.1×), Gaussian blur ($\sigma \in [0, 0.8]$), additive Gaussian noise ($\sigma \in [0, 0.01\times255]$), and brightness multiplication (between 0.8× and 1.2×). Each original image produced 20 augmented versions, significantly increasing the dataset's variation.

Face Detection and Cropping (YOLOv8)

The first stage in the pipeline is to localize faces in each video frame. The researchers use the YOLOv8 model, a variant of You Only Look Once version 8 (table 1). YOLOv8 was chosen due to its state-of-the-art performance in terms of detection speed and accuracy (Karakuş et al., 2023). Unlike traditional anchor-based detectors, YOLOv8 uses an anchor-free head, which reduces computational complexity and improves generalization (Yuan et al., 2024).

The system specifically utilizes the pre-trained *yolov8m-face.pt* model, which has been optimized for face detection. As implemented in the detect_faces function, this model only produces the bounding box coordinates (x1, y1, x2, y2) and a confidence score for each detection. This implementation does not use segmentation masks; instead, it focuses on the extraction of a rectangular bounding box-based Region of Interest (RoI).

The face cropping process is performed precisely using these bounding box coordinates. To ensure all facial features (context) are captured, the

system applies padding enhancement. As per the code, a padding of 15% of the bounding box width (w) and height (h) is added to each side (pad_x = int(0.15 * w)) before the frame is cropped. This process ensures that only the relevant facial features, along with their surrounding context, are passed to the feature extraction stage.

Figure 3. Real-Time Face Detection using YOLOv8

The system's detection optimization applies several filters to enhance detection quality. As defined by the parameters in the FaceDetector class, the filters used are confidence_threshold = 0.7 (only detections with a confidence >70% are processed) and min_face_size = 40 (faces smaller than 40x40 pixels are ignored).

The confidence_threshold = 0.7 was selected to balance between false positives (incorrectly detecting non-faces as faces) and false negatives (failing to detect actual faces), ensuring that only high-confidence detections proceed to the recognition stage. This threshold represents a strict yet practical confidence level that maintains detection reliability in real-world conditions.

The min_face_size = 40 parameter was determined based on the minimum facial pixel area required for reliable recognition from a 3-meter height, considering the camera's resolution limitations and the expected face-to-camera distance. Faces smaller than this threshold are typically too degraded for accurate feature extraction and are therefore filtered out to optimize processing efficiency.

Table 1. Yolov8m-face architecture

type	input	output	stride	Ouput size
BACKBONE LAYERS				
Conv Stem layar	3	48	2	480 x 480
Conv Downsample	3	48	2	240x 240
C2f Feature Extraction	96	96	1	240 x 240
Conv Downsample	96	192	2	120 x 120
Cf2 Feature Extraction	192	192	1	120 x 120
Conv Downsample	192	384	2	60 x 60
C2f Feature Extraction	384	384	1	60 x 60
Conv Donwsample	384	576	2	30 x 30
C2f Feature Extraction	576	576	1	30 x 30
SPPF Spatial Pyramid Pooling	576	576	1	30 x 30
Neck Layers				
Upsample	576	576	-	60 x 60
Concatenate Features	576 + 384	960	-	60 x 60
C2f Feature Fusion	960	384	-	60 x 60
Upsample	384	384	-	120 x 120
Concatenate Features	384 + 192	576	-	120 x 120
C2f Feature Fusion	576	192	-	120 x 120
Conv Downsample	192	192	-	60 x 60
Concatenate Feature	192 + 384	576	-	60 x 60
C2f Feature Fusion	576	384	-	60 x 60
Conv Downsample	384	384	-	30 x 30
Concatenate Feature	384 + 576	960	-	30 x 30
C2f Feature Fusion	960	576	-	30 x 30
Head Layers				
Detect Multi-scale detection head	P3 (192ch), P4 (384ch), P5 (576ch)	Bounding boxes + Confidence	-	-

Face Feature Extraction (ArcFace)

After the face is detected and cropped with padding, the facial image is processed using the consistent pre-processing pipeline (CLAHE, Denoising, Sharpening) as described in Section A. At the end of this pipeline, the quality-enhanced facial image is uniformly resized to 224x224 pixels.

This prepared image is then fed into the ArcFace facial recognition model. This model uses a Deep Convolutional Neural Network (DCNN) architecture with a ResNet-100 backbone to transform the facial image into a high-dimensional feature vector (embedding), specifically 512-D.

The advantage of ArcFace lies in its loss function, the Additive Angular Margin Loss. This function maximizes the angular distance between different identities (inter-class) and minimizes the intra-class distance within the feature space. The ArcFace loss function is defined in Equation (1):

$$L = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s\cdot(\cos(\theta_{yi}+m))}}{e^{s\cdot(\cos(\theta_{yi}+m))} + \sum_{j=1,j\neq y1}^{n} e^{s\cdot\cos(\theta_j)}} \quad (3)$$

Where :

- N is the number of samples (batch size).
- n is the number of classes (total employees).
- θ_{y_i} is the angle between the sample embedding i and the class center y_i
- m is the additive angular margin (typically m = 0.5)
- s is the feature scaling scalar (typically s = 64)

The use of this margin m effectively increases the model's discriminative power by forcing the model to create a stricter decision boundary between different classes. This margin results in tighter clustering for samples from the same class and greater separation between classes.

After extraction, each embedding vector is normalized using L2 normalization. This step is crucial to ensure all feature vectors lie on the unit hypersphere with the same magnitude, thus making distance comparisons (such as cosine similarity or the input to SVM) valid. This normalization is implemented explicitly in the system and is defined in Equation (2):

$$\hat{e} = \frac{e}{\|e\|_2} \quad (4)$$

Where $\|e\|_2$ is the L2 norm of embedding vector e, calculated as $\sqrt{\sum_{k=1}^{512} e_k^2}$.

Identity Classification (Support Vector Machine)

The 512-dimensional feature vectors generated by the ArcFace model are used as input to train a classifier. Although 1:1 face verification can be performed using cosine similarity, this approach is less efficient in the context of a multi-user attendance system. Therefore, this research utilizes a supervised classification approach that is faster and more scalable for real-time inference.

The face classification model was built using the Support Vector Machine (SVM) algorithm. SVM is a supervised classifier that works by finding the optimal hyperplane to separate data between classes with the maximum margin. Unlike the RBF kernel configuration, this research specifically implements the Polynomial (poly) kernel. This kernel was chosen for its ability to map non-linear embedding data into a higher-dimensional space where it can be effectively separated.

The SVM hyperparameters were adjusted for optimization to achieve a balance between generalization and classification performance. The configuration implemented in the system is: kernel = 'poly', C = 1.0 (regularization parameter), degree = 3 (degree for the polynomial kernel), gamma = 'scale' (kernel coefficient), and class_weight = 'balanced' (to handle potential data imbalance between classes).

The system also activates probability estimation (probability = True) to generate confidence scores, and sets a random state (random_state = 42) to ensure the reproducibility of training results. The model was trained offline using the embedding data, which had been split into an 70% training and 30% testing ratio (as defined in the 'build embeddings' stage). Mathematically, the SVM decision function (for an input x) is defined as:

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right) \quad (5)$$

Where :

- α_i is the koefisien Lagrange,
- y_i is the class label,
- b is the bias,
- $K(x_i, x)$ is the kernel function.

Fungsi kernel Polinomial yang digunakan didefinisikan sebagai:

$$K(x_i, x) = (\gamma(x_i \cdot x_j) + r)^d \quad (6)$$

Where γ is the gamma coefficient, r is the coefθ (default 0) dan d is the derajat (degree=3).

During the real-time inference stage, each extracted facial embedding is fed into the pre-trained SVM model. This model generates an identity prediction (employee name) along with probability scores for each class. The system implements a dual-threshold mechanism to enhance reliability and reduce false positives. The SVM Classifier Threshold is evaluated first, and a prediction is only accepted if its probability confidence score is above 0.65. If the SVM Similarity

Threshold is below 0.65, the system switches to a fallback mechanism. This mechanism calculates the cosine similarity between the input embedding and the embeddings stored in the database. The identity is accepted only if the similarity score exceeds 0.70. This threshold was determined through empirical testing on a validation set to optimize the trade-off between the True Acceptance Rate (TAR) and the False Acceptance Rate (FAR), ensuring a high rate of correct identifications while minimizing the acceptance of unknown or impostor faces. This approach ensures the system can maintain accuracy and stability under various lighting, pose, and facial expression conditions.

Testing Scenario and Evaluation Metrics

To comprehensively evaluate the system's recognition performance, we use a set of standard classification evaluation metrics commonly employed in computer vision research. The evaluation focuses on the performance of the SVM classifier in accurately identifying individuals.

1) Dataset Splitting

The set of embedding data extracted from the database (including 20 augmentations per source image) was divided using the Stratified Train-Test Split method. The split ratio used was 70% training data and 30% testing data. The use of stratification ensures that the proportion of each employee class (identity) is maintained proportionally in both the training and testing sets. A seed of 42 was used to ensure the reproducibility of the data splitting and model training results.

2) Detection Performance (YOLOv8)

In this research, the YOLOv8 detection model (yolov8m-face.pt) functions as a localization component (a tool) to provide input for the recognition pipeline. The quantitative evaluation focus of this study is on the classification (recognition) performance; therefore, detection accuracy metrics (such as mAP) were not calculated.

The only performance metric evaluated for the detection component is its computational performance (speed), measured as the Average Detection Time (ADT). This metric is calculated as the arithmetic mean of the frame processing time:

$$ADT = \frac{1}{N}\sum_{i=1}^{N}(T_{e_i} - T_{s_i}) \quad (7)$$

Where N is the number of frames measured, T_{e_i} is the end time and T_{s_i} start time. This result is presented in the Computational Performance Analysis section to validate the feasibility of the real-time system.

3) Recognition Performance (SVM)

The SVM classifier's performance was evaluated quantitatively on the test set (30% unseen data). Comprehensive metrics were calculated from the confusion matrix. As per the system's implementation, the calculated metrics are the weighted average to account for potential class imbalance. The primary metrics used include:

- Accuracy : The proportion of total correct predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

- Precision - Weighted Average: The model's ability to not label a negative sample as positive.

$$Precision_{weighted} = \sum_{i=1}^{n} w_i \cdot \frac{TP_i}{TP_i + FP_i} \quad (9)$$

- Recall (Perolehan) - Weighted Average: Kemampuan model untuk menemukan semua sampel positif.

$$Recall_{weighted} = \sum_{i=1}^{n} w_i \cdot \frac{TP_i}{TP_i + FN_i} \quad (10)$$

- F1-Score - Weighted Average: The harmonic mean of Precision and Recall.

$$F1_{weighted} = \sum_{i=1}^{n} w_i \cdot 2 \cdot \frac{Precision_i \cdot Recall_i}{Precision_i + Recall_i} \quad (11)$$

Where n is the number of classes, w_i is the weight (proportion) of class i relative to the total samples, TP is True Positive, FP is False Positive, TN is True Negative, and FN is False Negative.

4) Confusion Matrix Analysis

A $n \times n$ confusion matrix C (where θ is the number of classes) was analyzed to identify classes (employees) that were frequently misclassified, detect confusion patterns between classes with similar visual features, and calculate per-class precision and recall individually.

5) Ablation Study: SVM vs. Cosine Similarity

To validate the choice of classification architecture, we conducted an ablation study. The hybrid architecture implemented in the system allows for a direct comparison between two classification approaches:

- Proposed System (SVM Classifier): Uses the pre-trained SVM classifier (Polynomial kernel). Predictions are accepted if the probability confidence score exceeds 0.60.
- Baseline System (Cosine Similarity): Uses a 1:1 cosine similarity comparison. The identity is accepted if the similarity score exceeds 0.65.

The baseline system (Cosine Similarity) also functions as a fallback mechanism in the proposed system when the SVM confidence falls below the 0.60 threshold. The cosine similarity between two d-dimensional (512) embeddings A and B is defined as:

$$Similarity(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{d} A_i B_i}{\sqrt{\sum_{i=1}^{d} A_i^2}\sqrt{\sum_{i=1}^{d} B_i^2}} \quad (12)$$

6) Computational Performance Analysis

To evaluate the feasibility of real-time implementation, the system is designed to measure several computational performance metrics granularly during training and inference:

- Total Training Time: The total time (in seconds) required to train the SVM classifier offline on 70% of the training set.
- Detection Time: The average time (in milliseconds) required by the detection module (YOLOv8) to process a single frame and return facial bounding boxes.
- Recognition Inference Time**:** The average time (in milliseconds) required by the recognition module for a single face. This includes the combined overhead of facial crop pre-processing, ArcFace embedding feature extraction, and classification (SVM prediction) or cosine similarity calculation.

Table 2. Hardware and tools specifications

Components	Specifications
Processor	Intel core i5
RAM	16 GB
GPU	NVIDIA GeForce MX550
OS	Windows 11
Python	3.10.8
PyTorch	2.5.1+cu121
CUDA	12.1
Yolov8	YOLOv8m-face
DeepFace	0.0.79
Scikit-learn	1.3.0

RESULTS AND DISCUSSION

This section presents the experimental evaluation results of the proposed pipeline. Testing was conducted on the specified hardware (Table 2) using an internally collected dataset from the hospital/institutional environment.interpretation of these results are required before the discussion.

Dataset Characteristics

Table 3. Database characteristics

Metrik	Training Set	Testing Set	Total
Number of employees	5	5	5
original images	21	9	30
Total after augmentation	441	189	630
Average images per employee	6	6	6
Average images per employee	2592 x 1944 pixels (5mp)	-	-
Average faces detected	23.9%	-	-
Size range	>= 40 pixels	-	-
Augmentation Factor	20 x per image	-	-

This dataset in table 3, consisting of 30 original images, was augmented using 20 augmentations per image, resulting in a total of 630 samples. Following the 70/30 split ratio, the data was divided into 441 training samples (70%) and 189 testing samples (30%). The class distribution visualization in Figure 4 shows that the training dataset used is highly balanced, with the number of samples per class ranging from 88 to 89.

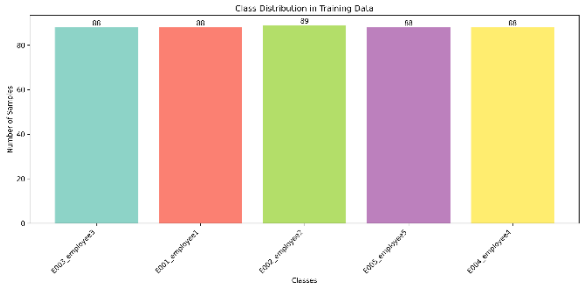


Figure 4. Class Distribution in the Training Set. This bar chart illustrates the total number of samples for each of the five employee classes after the augmentation process.

To analyze the separability of features extracted by ArcFace, dimensionality reduction was performed using Principal Component Analysis (PCA) from the 512-D feature space down to 2-D (Figure 5). The PCA visualization shows that although some identity clusters appear distinct, there is significant overlap among other classes.

In particular, the clusters for E002_employee2 (purple) and E005_employee5

P-ISSN: 2656-1743 | E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.465
JURNAL RISET INFORMATIKA
Vol. 8, No. 1. December 2025
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

(light green), as well as E001_employee1 (red) and E005_employee5, exhibit high spatial proximity. This overlap validates the methodological decision to use a non-linear classifier, such as an SVM with a Polynomial kernel, which is more capable of modeling complex decision boundaries.

Figure 5. PCA Projection of the Training Data Feature Space. The plot shows the 2D distribution of the five identity classes after dimensionality reduction.

Classification Model Performance (SVM)

The SVM model's performance was quantitatively evaluated on the 30% test set (189 samples). The aggregate (weighted average) performance metrics are presented in Table 4.

Table 4. System evaluation metrics

Metric	Score
accuracy	0.937
Precision	0.938
recall	0.937
F1-Score	0.935

The system achieved excellent performance with 93.7% accuracy and an F1-Score of 0.935, indicating a strong balance between precision and recall. A more in-depth error analysis is presented in the confusion matrix (Figure 6). This matrix confirms the high performance, where the E001_employee1 and E004_employee4 classes achieved 100% accuracy. However, this matrix also clearly identifies the system's primary source of error:

- The E005_employee5 class had the lowest performance, with a per-class accuracy of only 70.9%.
- A significant misclassification pattern was identified where 15.8% of E005_employee5 samples were misclassified as E002_employee2.

This error pattern correlates directly with the cluster overlap observations in the PCA analysis (Figure 5) and the visual similarity between the individuals (Figure 6).

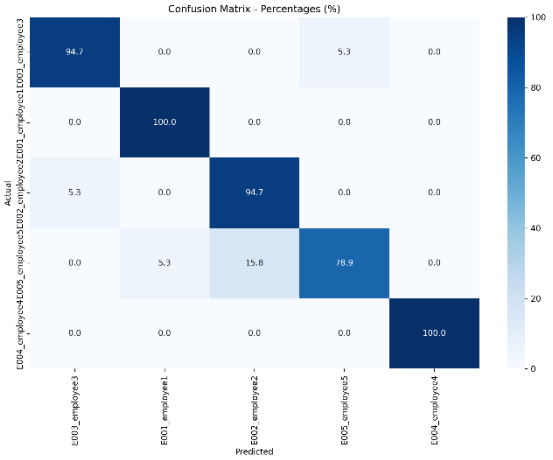
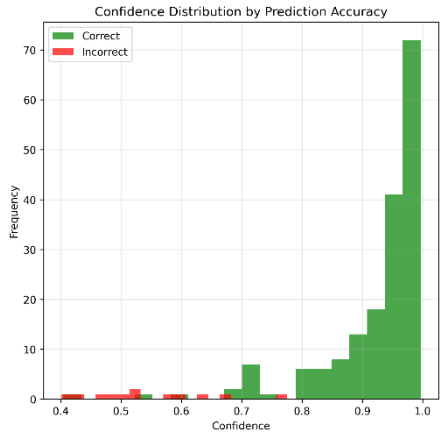


Figure 6. SVM Performance Confusion Matrix Analysis (Percentage). The main diagonal shows the per-class accuracy.

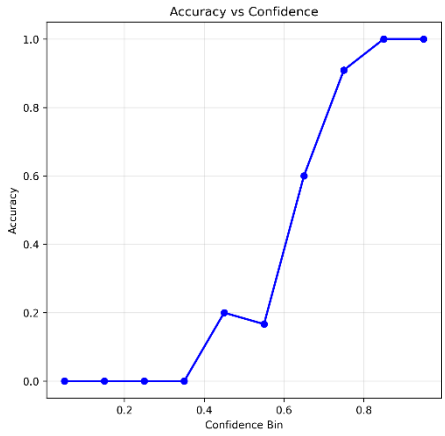
Reliability and Confidence Analysis

To validate system reliability, a correlation analysis was conducted between the SVM confidence scores and the actual prediction accuracy (Figure 7). The distribution plot (a) shows that the majority of Correct predictions (green) have high confidence scores (> 0.90), whereas Incorrect predictions (red) tend to have low confidence scores.

The "Accuracy vs. Confidence" plot (b) quantifies this relationship. A strong positive correlation is observed: the higher the confidence score, the higher the accuracy. Crucially, predictions with a confidence score of 0.80 or higher achieve 100% accuracy. This finding validates the use of a classifier threshold of 0.60. At the 0.60 bin, the prediction accuracy is approximately 60%, making it a reasonable threshold to balance the acceptance of correct predictions and the rejection of incorrect ones.



(a)



(b)

Figure 7. SVM Prediction Confidence Analysis. (a) Frequency distribution of confidence scores for Correct vs. Incorrect predictions. (b) Plot of Accuracy vs. Confidence Bin.

Dataset Characteristics

Real-time performance evaluation was conducted during a testing session that processed 117 frames and detected 28 faces. The computational performance summary is presented in Table 5.

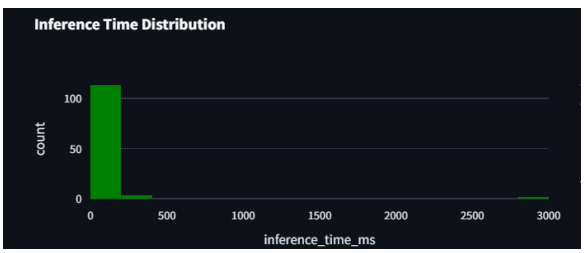
Table 5. Trial session computational performance summary

Metrik	Nilai
Waktu Inferensi Rata-rata (Avg)	139.8 ms
Waktu Inferensi Maksimum (Max)	2981.2 ms
Waktu Inferensi Minimum (Min)	87.1 ms
Estimasi FPS (Frames Per Second)	7.2
Total Wajah Terdeteksi	28
Total Frame Diproses	117

The system achieved an average FPS of 7.2, which is adequate for real-time attendance applications. The average inference time (encompassing detection, preprocessing, and recognition) was 139.8 ms. The distribution of inference times, as shown in Figure 8, reveals important characteristics about system behavior in practical deployment. Most frames (over 85%) were processed within 250 ms, indicating consistent performance under normal conditions. This distribution directly impacts user experience: while 7.2 FPS may appear laggy for video display, it provides sufficient temporal resolution for attendance logging where each face needs to be identified only once per entry event.

The outlier with maximum inference time of 2981.2 ms (approximately 3 seconds) was identified during initial system warm-up. This spike can be attributed to several factors: (1) initial model loading and CUDA kernel compilation when processing the first frames, (2) memory allocation overhead for the first batch of detections, and (3) potential CPU/GPU synchronization delays during pipeline initialization. Such initialization overhead is common in deep learning systems and does not significantly impact continuous operation once the system stabilizes.

The time-per-frame analysis in Figure 9 further confirms that this was an isolated incident, with subsequent frames returning to normal processing times. This distribution pattern has practical implications: in a real deployment, the system can maintain consistent performance once initialized, with occasional variability due to factors like changing numbers of faces per frame or varying computational load from other system processes. The 7.2 FPS throughput, while modest, represents approximately 7 identification opportunities per second—more than sufficient for attendance systems where individuals typically pause briefly during entry.



(a)

P-ISSN: 2656-1743 | E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.465
JURNAL RISET INFORMATIKA
Vol. 8, No. 1. December 2025
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

(b)

Figure 8. System Inference Time Statistics Visualization (ms). (a) Histogram showing the frequency distribution of per-frame processing time. (b) Box plot illustrating the quartile distribution and identifying data outliers.

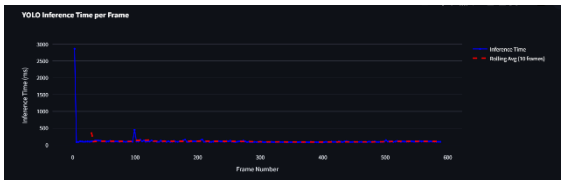
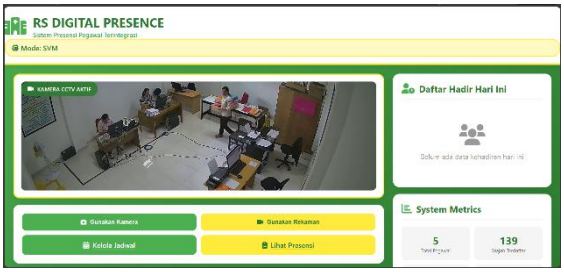


Figure 9. YOLO Inference Time per Frame. The plot shows the actual inference time (blue) for each frame and the moving average (red, 10 frames), which identifies an outlier (spike) in the initial frames.

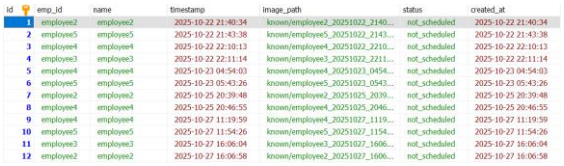
Operational System Validation and Attendance Pipeline

Beyond quantitative classification and speed metrics, the system's functionality as a complete, operational attendance management solution was validated through its end-to-end pipeline. As illustrated in the system dashboard (Figure 10a), the application provides a comprehensive user interface for managing the attendance process. The interface displays real-time system metrics (e.g., total employees, registered faces), offers controls for camera and recording management, schedule configuration, and direct access to attendance logs, confirming its readiness for practical deployment.

The core attendance logging functionality was successfully tested in a real-world simulation. The system's backend database automatically records each recognition event with critical metadata. A sample excerpt from the operational attendance log is presented in Figure 10b. It includes fields such as employee ID (emp_id), name, the precise timestamp of recognition, the path to the captured face image (image_path), attendance status, and the record's created_at timestamp.



(a)



id	emp_id	name	timestamp	image_path	status	created_at
1	employee2	employee2	2025-10-22 21:40:34	known/employee2_20251022_2140...	not_scheduled	2025-10-22 21:40:34
2	employee5	employee5	2025-10-22 21:43:38	known/employee5_20251022_2143...	not_scheduled	2025-10-22 21:43:38
3	employee4	employee4	2025-10-22 22:10:13	known/employee4_20251022_2210...	not_scheduled	2025-10-22 22:10:13
4	employee3	employee3	2025-10-22 22:11:14	known/employee3_20251022_2211...	not_scheduled	2025-10-22 22:11:14
5	employee4	employee4	2025-10-23 04:54:03	known/employee4_20251023_0454...	not_scheduled	2025-10-23 04:54:03
6	employee5	employee5	2025-10-23 05:43:26	known/employee5_20251023_0543...	not_scheduled	2025-10-23 05:43:26
7	employee2	employee2	2025-10-25 20:39:48	known/employee2_20251025_2039...	not_scheduled	2025-10-25 20:39:48
8	employee4	employee4	2025-10-25 20:46:55	known/employee4_20251025_2046...	not_scheduled	2025-10-25 20:46:55
9	employee4	employee4	2025-10-27 11:19:59	known/employee4_20251027_1119...	not_scheduled	2025-10-27 11:19:59
10	employee5	employee5	2025-10-27 11:54:26	known/employee5_20251027_1154...	not_scheduled	2025-10-27 11:54:26
11	employee3	employee3	2025-10-27 16:06:04	known/employee3_20251027_1606...	not_scheduled	2025-10-27 16:06:04
12	employee2	employee2	2025-10-27 16:06:58	known/employee2_20251027_1606...	not_scheduled	2025-10-27 16:06:58

(b)

Figure 10. (a) Operational System Dashboard. The main interface of the implemented attendance system, showing live metrics and management controls. (b) Sample Database Log. A view of the structured attendance records stored by the system, demonstrating successful end-to-end data pipeline execution.

The log demonstrates the system's capability to seamlessly integrate the AI pipeline (detection → recognition) with backend business logic and database operations. For instance, entries show the status 'not_scheduled', indicating the system's ability to not only identify an employee but also cross-reference their identity with a work schedule—a crucial feature for valid attendance recording. This integration from CCTV video feed to structured database record, accessible via a dedicated UI, substantiates the claim of a fully functional, automated, and end-to-end attendance system.

CONCLUSIONS AND SUGGESTIONS

Conclusion

This research has successfully designed, implemented, and evaluated an automated attendance system that combines YOLOv8m for face detection, ArcFace for feature extraction, and a Polynomial kernel SVM for classification. The system demonstrated high classification performance with 93.7% accuracy and an F1-Score of 0.935 on a held-out test set. Furthermore, operational validation confirms its end-to-end functionality. The system successfully processes live video input, executes the AI recognition pipeline, makes attendance decisions based on schedules, logs structured records to a database, and presents data through a functional user interface (Figure 10). This complete integration—

from perception to data persistence—validates its readiness as a practical replacement for manual attendance methods. Computationally, the system achieves a throughput of 7.2 FPS, which is sufficient for the attendance use case where individuals are logged upon entry.

Suggestion

Confusion matrix and PCA analysis successfully identified the system's main challenge: feature overlap between E005_employee5 and E002_employee2, which was the primary cause of misclassification. Furthermore, its real-time performance on the tested hardware is limited, with an estimated throughput of 7.2 FPS. It should be noted that 7.2 FPS is not sufficient for visually smooth video playback and will appear "laggy".

REFERENCES

Adelia, N., Munthe, A. A., & Masrizal. (2021). Sistem Informasi Reservasi Hotel Rantauprapat Berbasis Web Dengan Framework. *Development Information System (JoSDIS)*, *1*. https://doi.org/https://doi.org/10.36987/josdis.v1i1.2197

Adim, M. S., & Nurhidayat, A. I. (2023). *Pembuatan Sistem Presensi Berbasis Pengenalan Wajah Dengan Metode You Only Look Once Version 8 (Studi Kasus: Event Organizer SHAF Management)*. https://ejournal.unesa.ac.id/index.php/jurnal-manajemen-informatika/article/view/63001/47815

Alzu'bi, A., Albalas, F., AL-Hadhrami, T., Younis, L. B., & Bashayreh, A. (2021). Masked Face Recognition Using Deep Learning: A Review. *Electronics*, *10*(21), 2666. https://doi.org/10.3390/electronics10212666

Arishadilah, R. (2024). *Perancangan Sistem Presensi Guru Berbasis Pengenalan Wajah Menggunakan Metode Klasifikasi Haar Cascade di MTS Al Fakhriyyah* [Thesis]. UNIVERSITAS SATYA NEGARA INDONESIA.

Armiady, D. (2021). Absensi Kehadiran Menggunakan Kamera Pengawas Berbasis Teknologi Computer Vision. *JURNAL TIKA*, *6*(02), 140–146. https://doi.org/10.51179/tika.v6i02.541

Dony, D., & Lubis, C. (2025). Deteksi YOLOv8 dan Pengenalan Wajah Menggunakan RESNET50 Pada Gereja. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *12*(1). https://doi.org/10.35957/jatisi.v12i1.9757

Dumbere, M., Khan, R., Satpute, A., Pathan, S., & Ramteke, S. (2024). Smart CCTV (Face Recognition Attendance). *IJARSCT*, *4*(4). https://doi.org/10.48175/IJARSCT-18331

Fan, Y., Luo, Y., & Chen, X. (2021). Research on Face Recognition Technology Based on Improved YOLO Deep Convolution Neural Network. *Journal of Physics: Conference Series*, *1982*(1). https://doi.org/10.1088/1742-6596/1982/1/012010

Hasan, & Hardjianto, M. (2024). Pengenalan Wajah Secara Realtime Menggunakan Adaboost Viola-Jones dan 2D DWT-PCA dengan Struktur Index KNN-KD Tree. *Decode: Jurnal Pendidikan Teknologi Informasi*, *4*(1), 154–166. https://doi.org/10.51454/decode.v4i1.300

Karakuş, S., Kaya, M., & Tuncer, S. A. (2023). Real-Time Detection and Identification of Suspects in Forensic Imagery Using Advanced YOLOv8 Object Recognition Models. *Traitement Du Signal*, *40*(5), 2029–2039. https://doi.org/10.18280/ts.400521

Listyoningrum, K. I., Fenida, D. Y., & Hamidi, N. (2023). Inovasi Berkelanjutan dalam Bisnis: Manfaatkan Flowchart untuk Mengoptimalkan Nilai Limbah Perusahaan. *Jurnal Informasi Pengabdian Masyarakat*, *1*(4), 100–112. https://doi.org/10.47861/jipm-nalanda.v1i4.552

Mahalwal, L., Gulati, J., Duhan, L., & Mishra, A. D. (2020). Smart Display Board With CCTV For Attendance. *India International Journal of Technical Research*. https://doi.org/10.30780/specialissue-ICACCG2020/0033

Marappan, S., Kuppuswamy, P., John, R., & Shanmugavadivu, N. (2021). *Human Detection in Still Images Using Hog with SVM: A Novel Approach* (pp. 385–397). https://doi.org/10.1007/978-3-030-65407-8_33

Mustafa Abdullah, D., & Mohsin Abdulazeez, A. (2021). Machine Learning Applications based on SVM Classification A Review. *Qubahan Academic Journal*, *1*(2), 81–90. https://doi.org/10.48161/qaj.v1n2a50

Nawawi, M. (2023). *Sistem Presensi Sekolah SMK Queen Al-Falah Menggunakan Face Recognition* [Thesis, Universitas Nusantara PGRI Kediri]. https://repository.unpkediri.ac.id/12027/3/RAMA_55201_19103020098_0711018102_0729098903_01_front_ref.pdf

Nguyen, T.-A., Tran-Thi, T.-Q., Bui, D.-H., & Tran, X.-T. (2023). FPGA-Based Human Detection System using HOG-SVM Algorithm. *2023 International Conference on Advanced Technologies for Communications (ATC)*, 72–77. https://doi.org/10.1109/ATC58710.2023.10318871

Nurlita, B. W., Sri Winarno, Adhitya Nugraha, Almas Najiib Imam Muttaqin, Yasmin Zarifa, Pramesya Mutia Salsabila, & Ghina Fairuz Mumtaz7. (2024). Comparison of ArcFace and Dlib Performance in Face Recognition with Detection Using YOLOv8. *INOVTEK Polbeng - Seri Informatika*, *9*(2), 890–903. https://doi.org/10.35314/3jy3dy73

Ren, Z., Liu, X., Xu, J., Zhang, Y., & Fang, M. (2025). LittleFaceNet: A Small-Sized Face Recognition Method Based on RetinaFace and AdaFace. *Journal of Imaging*, *11*(1), 24. https://doi.org/10.3390/jimaging11010024

Romli, I. (2021). Penerapan Data Mining Menggunakan Algoritma K-Means Untuk Klasifikasi Penyakit Ispa. *Indonesian Journal of Business Intelligence (IJUBI)*, *4*(1), 10. https://doi.org/10.21927/ijubi.v4i1.1727

Sim, J. H., Ma'muriya, N., & Yulianto, A. (2024). Evaluating YOLOv5 and YOLOv8: Advancements in Human Detection. *Journal of Information Systems and Informatics*, *6*(4), 2999–3015. https://doi.org/10.51519/journalisi.v6i4.944

Sydor, A., Balazh, D., Vitrovyi, Yu., Kapshii, O., Karpin, O., & Maksymyuk, T. (2024). Research On The State-Of-The-Art Deep Learning Based Models For Face Detection And Recognition. *Information and Communication Technologies, Electronic Engineering*, *4*(2), 49–59. https://doi.org/10.23939/ictee2024.02.049

Terven, J., & Cordova-Esparza, D. (2023). *A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS*. https://doi.org/10.3390/make5040083

Tofir, S., Leppang, I., & Hamzah, M. A. (2020). Perancangan Sistem Informasi Pada Dinas Pendidikan Kota Palopo Berbasis Web. *Jurnal Health Sains*, *1*(6), 762–772. https://doi.org/10.46799/jsa.v1i6.107

Valero-Carreras, D., Alcaraz, J., & Landete, M. (2023). Comparing two SVM models through different metrics based on the confusion matrix. *Computers and Operations Research*, *152*. https://doi.org/10.1016/j.cor.2022.106131

Walangitan, J., Sompie, S. R. U. A., & Najoan, X. B. N. (2024). Sistem Absensi Pengenalan Wajah Bermasker. *Jurnal Teknik Informatika*, *19*(01), 21–30. https://doi.org/10.35793/jti.v19i01.51327

Yisihak, H. M., & Li, L. (2024). Advanced Face Detection with YOLOv8: Implementation and Integration into AI Modules. *OALib*, *11*(11), 1–19. https://doi.org/10.4236/oalib.1112474

Yu, J., & Zhang, W. (2021). Face Mask Wearing Detection Algorithm Based on Improved YOLO-v4. *Sensors*, *21*(9), 3263. https://doi.org/10.3390/s21093263

Yuan, Z., Shao, P., Li, J., Wang, Y., Zhu, Z., Qiu, W., Chen, B., Tang, Y., & Han, A. (2024). YOLOv8-ACU: improved YOLOv8-pose for facial acupoint detection. *Frontiers in Neurorobotics*, *18*. https://doi.org/10.3389/fnbot.2024.1355857

Zhang, X., Xuan, C., Xue, J., Chen, B., & Ma, Y. (2023). LSR-YOLO: A High-Precision, Lightweight Model for Sheep Face Recognition on the Mobile End. *Animals*, *13*(11). https://doi.org/10.3390/ani13111824