Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.5, October 2025, pp. 2049~2058
e-ISSN: 2548-6861

Face Recognition Using MTCNN Face Detection, ResNetV1 Feature Embeddings, and SVM Classification

Ivan Putra Pratama[1]*, Novita Kurnia Ningrum [2]*
* Teknik Informatika, Universitas Dian Nuswantoro
iivanpratama16@gmail.com [1], novita.kn@dsn.dinus.ac.id [2],

Article Info

Article history:

Received 2025-08-03
Revised 2025-09-08
Accepted 2025-09-10

Keyword:

Face Recognition,
MTCNN,
ResNetV1,
Support Vector Machine,
Deep Learning

ABSTRACT

Face recognition has become an essential component of modern security and authentication systems, yet its effectiveness is often challenged by limited datasets, class imbalance, variations in facial poses, lighting conditions, and image resolutions. This study proposes a face recognition pipeline that integrates Multi-task Cascaded Convolutional Networks (MTCNN) for face detection, Residual Network V1 (ResNetV1) for feature extraction, and Support Vector Machine (SVM) for classification. Unlike previous works that rely on large-scale datasets and end-to-end deep learning models, this study emphasizes the effectiveness of the pipeline under constrained data conditions, using 856 images across 191 classes with highly imbalanced distribution. Experimental results show that MTCNN successfully detected 97.1% of faces, while ResNetV1 produced 512-dimensional embeddings that formed well-separated clusters validated by clustering metrics (Silhouette Score = 0.578, Davies-Bouldin Index = 0.566). The SVM classifier achieved 92.9% accuracy, with macro-average precision, recall, and F1-scores of 0.89, 0.92, and 0.89 respectively, significantly outperforming a baseline k-Nearest Neighbor (k-NN) model that only reached 63.9% accuracy. These findings highlight the novelty of this study: demonstrating that a lightweight yet robust pipeline can deliver reliable recognition performance even in small, imbalanced datasets, making it suitable for real-world scenarios where large-scale training data are not available.

I. INTRODUCTION

Face Recognition systems are widely used by various technology developments, especially in security systems and attendance recording. Various constraints, such as varying photo angles, image resolutions, and lighting conditions, require a robust system even under various conditions

Various experimental studies have been conducted to tackle problems in face recognition, for example using Haar Cascade Classifier and Convolutional Neural Network with 10 datasets containing images, achieving accuracy up to 98,94% and average computation time of 0.5s [1]. These methods, however, have certain drawbacks. They can only recognize previously recorded faces in their original state, and when significant changes occur—such as those caused by plastic surgery—the methods often fail, treating the altered face as a completely new identity.

The MTCNN algorithm was selected due to its three-layer CNN architecture, which enables the detection of finer facial details. Moreover, it outperforms Haar Cascade Classifier due to its ability to handle variations in photo angles, enabling MTCNN to detect faces in a wider case compared to Haar Cascade Classifier [2].

However, the Viola-Jones Algorithm has been widely adopted for face recognition systems, studies have shown that Viola-Jones has limited ability under illumination changes, non-frontal faces poses, and low-resolution images. Viola-Jones Algorithm often results in false positive results under extreme poses or partial occlusion. These limitations strengthen the need of deep learning algorithms such as MTCNN to handle variations in pose or lighting [3].

MTCNN's capability is utilized to detect and mark face positions, allowing the model to focus on learning the important features while ignoring the noise in the photos.

Important face features are extracted so that the model can distinguish one label from another, and then learned by the Residual Network V1 algorithm [4].

Residual Network V1 addresses the vanishing gradient problem through shortcut connections, enabling deeper networks to learn rich face feature representations. This structure allows the network to be trained effectively even in deepest layers, allowing it to learn rich feature representations on every face. ResNetV1 generates embedding that captures low-level elements such as edges, corners, textures, and basic patterns, mid-level elements such as eyes, nose, mouth, and high-level elements represent abstract information such as various lighting, pose, and expression [5]. Embeddings typically represented as high-dimensional vectors (e.g. 128D or 512D), to represent essential specific features.

Support Vector Machine (SVM) are suited very well for classifying embedding results. First, SVM handles high-dimensional features effectively [6], it is ideal to process ResNetV1's embeddings. Second, SVM works well with feature extraction algorithms as it can separate important features. ResNetV1 as a feature extractor and integrates Support Vector Machine (SVM) for classification, providing an efficient approach for robust face recognition [6]

Integrating MTCNN, ResNetV1, and SVM forms a robust recognition pipeline. Unlike previous approaches that rely on traditional methods such as Haar Cascade, this study proposes a comprehensive face recognition pipeline using MTCNN for accurate face detection, ResNetV1 for feature extraction, and SVM for classification. ResNetV1 can capture important face features, SVM works well with high dimensional features, classifies the embeddings with high accuracy, as demonstrated in the previous study [6][7].

Therefore, the goal of this study is to implement a robust face recognition system combining MTCNN, ResNetV1, and SVM, and evaluate its performance accuracy and robustness.

Another challenge that motivates this study is the limited dataset scenario, where each class contains only a few samples and the distribution is highly imbalanced. Such conditions are common in real-world applications, where collecting large-scale balanced datasets is often impractical. Therefore, there is an urgent need for a recognition pipeline that remains effective under these constraints.

The novelty of this study lies in evaluating the effectiveness of integrating MTCNN, ResNetV1, and SVM in a limited dataset scenario with a high number of classes but few samples per class. Unlike previous works that mainly rely on large-scale datasets and end-to-end deep learning models, this study demonstrates that the proposed pipeline achieves robust recognition performance under constrained data conditions.

II. METHODOLOGY

The process of the proposed face recognition system follows a structured pipeline that ensures accuracy at each stage. It begins with dataset preparation, where 856 images across 191 classes are organized. The next step involves face detection using MTCNN, which is capable of handling variations in angle, scale, and illumination, ensuring that facial regions are accurately localized even under challenging conditions. Once the faces are detected, feature extraction is performed using ResNetV1, a deep residual network that generates 512-dimensional embeddings representing unique and discriminative characteristics of each subject. These embeddings serve as compact yet powerful representations of the input images. Finally, the classification stage is carried out using an SVM classifier, which leverages the extracted embeddings to distinguish between subjects and make the final recognition decision.

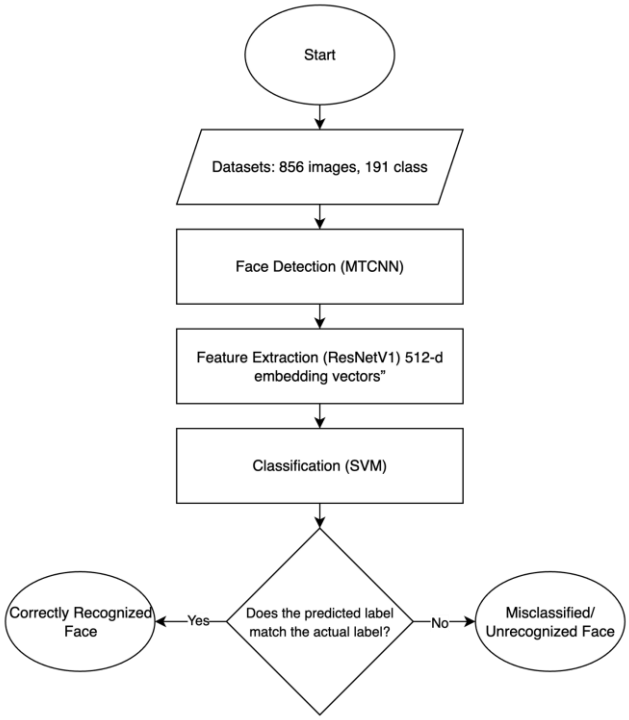

Figure 1. Overall Pipeline of The Face Recognition System.

Figure 1 illustrates rhe pipeline consists of four main stages: dataset preparation, face detection with MTCNN, feature extraction with ResNetV1, and classification using SVM. The process ensures that each face is accurately detected, represented, and classified into its corresponding label.

A. Datasets

The dataset used in this study was provided by a Computer Vision lecturer for academic purposes and consists of 191 classes with a total of 856 images. Among them, 520 images were allocated for training and 336 for testing, corresponding to a 61%–39% split. The split was predefined by the lecturer to ensure that the testing set contained facial images with different variations compared to the training set, thereby increasing the difficulty of the recognition task. On average, each class contains only 3 to 6 images, which leads to a highly imbalanced distribution. This imbalance presents a significant challenge because the model must generalize across classes with very limited intra-class variation.

All images were originally in PPM format and were converted to JPG for compatibility. To standardize the input and facilitate the learning process, each image was resized to 500×500 pixels, as this resolution was sufficient to cover the facial region in nearly all cases. The resized images were then processed with MTCNN to localize the face area, ensuring consistent alignment and maximizing the quality of feature extraction.

No data augmentation was applied in this study in order to preserve the originality and authenticity of the dataset. While this decision maintains the natural characteristics of the data, it also amplifies the impact of class imbalance, which is reflected in several misclassifications. This limitation, however, highlights the performance stability of the proposed pipeline, as demonstrated by its competitive results under highly constrained and imbalanced data conditions. Future work may incorporate augmentation strategies to further improve performance on classes with very limited samples. Figure 2 presents examples from the dataset, showing variations in facial expressions and poses.

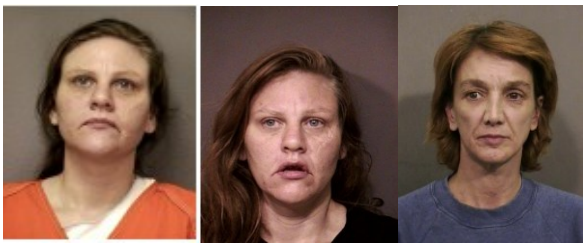

Figure 2. Images Example from Dataset

Figure 2 presents sample images from the dataset, illustrating variation in facial expression and face positions. Although each class contains only a limited number of samples, these variations provide diverse learning examples that enhance the model's ability to generalize.

B. Face Detection (MTCNN)

In this study, the MTCNN (Multi Task Convolutional Neural Network) was selected for face detection because of the efficiency and ability to handle various images with different challenges, such as varying photo angle, resolution and lighting conditions [2]. MTCNN operates three convolutional neural network layers, such as P-Net (Proposal Network), R-Net (Refine Network), and O-Net (Output Network). Each network performs specific tasks to progressively refine the detection process [8].

TABLE I
MTCNN PROCESS DESCRIPTION

Network Layers	Process
P-Net	Scans the image at multiple scales to propose candidate face regions [9].
R-Net	Refines the face proposals from P-Net by eliminating false positives and improving localization [9].
O-Net	Detects face landmarks (eyes, nose, mouth) and provides final bounding box adjustments [9].

Based on Table I, the MTCNN face detection process is through three sequential stages. The P-Net acts as a proposal generator, producing initial candidate regions at multiple scales to ensure that small and large faces can be captured.

These proposals are then refined by the R-Net, which reduces false positives and improves bounding box accuracy. Finally, the O-Net provides the most accurate detection by adjusting the bounding box further and identifying key facial landmarks such as the eyes, nose, and mouth [9].

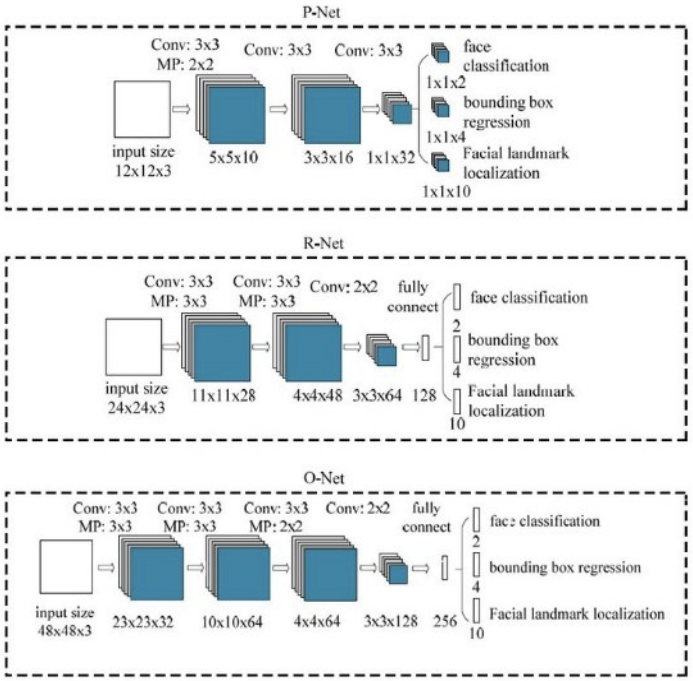

Figure 3. MTCNN three layers process [9].

Figure 3 illustrates three layers that work sequentially to process the images and progressively refine the detection results. The P-Net first generates candidate face regions across multiple scales, ensuring that both small and large faces are captured. These candidates are then passed to the R-Net, which eliminates false positives and improves bounding box accuracy through further refinement. Finally, the O-Net performs the most precise detection by adjusting bounding boxes and identifying facial landmarks such as the eyes, nose, and mouth. The MTCNN provides tuning parameters to handle specific challenges in different datasets. This parameter allows researchers to fine tune the detection to maximize the performance depending on what dataset they used.

From Table 2, it can be seen that the MTCNN implementation in this study used its default configuration without additional tuning. The parameters, such as *min_face_size* of 20 pixels and a *scale_factor* of 0.709, were applied to ensure balanced detection performance across images.

TABLE II
THE DEFAULT CONFIGURATION WAS APPLIED WITHOUT ADDITIONAL TUNING.

MTCNN Parameters	Value	Description
min_face_size	20 pixels	defining the minimum size of faces to be detected
scale_factor	0.709	controlling the scaling factor for the image pyramid.
threshold_pnet	0.6	Confidence threshold for P-Net
threshold_rnet	0.7	Confidence threshold for R-Net
threshold_onet	0.7	Confidence threshold for O-Net

Meanwhile, the confidence thresholds for P-Net, R-Net, and O-Net were set at 0.6, 0.7, and 0.7 respectively, which represent the standard values commonly used to achieve reliable face detection results.

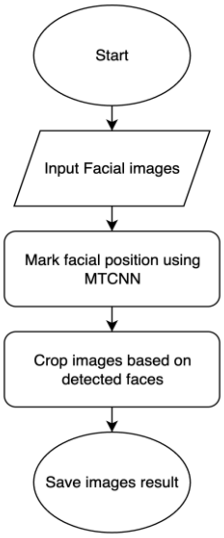

Figure 4. Preprocessing flowchart for face detection and alignment using MTCNN.

Figure 4 illustrates the preprocessing stage of this study, where MTCNN was implemented to detect faces in every image and mark the facial position with bounding boxes to maximize feature extraction. The detected faces were then cropped and aligned to ensure consistency in orientation and scale across the dataset. This preprocessing step is essential, as it reduces background noise and standardizes the input images, thereby improving the quality of the subsequent feature extraction process [10].

C. Feature Extraction (ResNetV1)

In this stage, the ResNetV1 was employed as the feature extractor to transform face images into numerical representations. Several studies have shown that ResNetV1 pretrained on large datasets such as VGGFace2, Casia-Webface [11]. Moreover, recent works have shown that improvements in residual networks enhance training stability and recognition accuracy. Residual blocks with refined shortcut connections yield faster convergence across image and video recognition [12], and comparative studies resulting that ResNet have better results compared to other lightweight models in terms of accuracy [13].

The output of the feature extractor is a fixed length embedding vector of 512 dimensions that encodes face characteristics [14]. These embeddings happened in a high-dimensional space, where the distance between vectors represent similarity between faces.

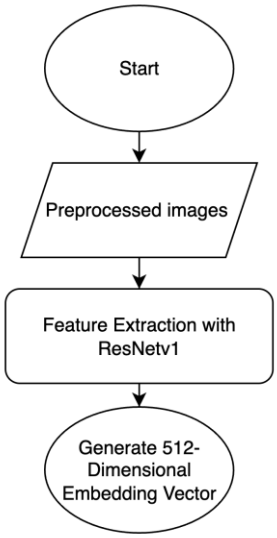

Figure 5. Flowchart Feature Extraction using ResNetV1

Figure 5 illustrates the feature extraction process begins with the preprocessed facial images, which are then passed into the ResNetV1 model. ResNetV1 extracts deep feature representations from each image, resulting in a 512-dimensional embedding vector. These embeddings capture the unique facial characteristics of each subject and are later used for classification, ensuring that the model can effectively distinguish between different individuals.

D. Classification (SVM)

Support Vector Machine (SVM) was used as the classifier to differentiate between individuals based on the feature embedding produced by the ResNetV1 model in the Feature Extraction step, the classifier trained using future extraction's 512-dimensional embedding feature.

SVM can be adjusted with various hyperparameters to suit the dataset and objective of the study [20]. SVM has been used in face recognition tasks because of its robustness to handle high dimensional data from Feature Extraction models such as ResNet [15].

In this study, the linear kernel was employed because the 512-dimensional embedding feature was considered that it is well separated, and the linear decision boundary is enough for classification.

Table III presents the parameters of the Support Vector Machine (SVM) used in this study. A linear kernel was selected because the 512-dimensional embeddings generated by ResNetV1 are assumed to be linearly separable in the feature space, making a linear decision boundary sufficient. The regularization parameter (C) was set to 1.0, which is the

default value in scikit-learn. This parameter balances the trade-off between achieving a wider margin and minimizing misclassification errors, ensuring that the model generalizes well without overfitting. For handling multi-class classification, the one-vs-one decision function shape was applied, where SVM constructs binary classifiers for each pair of classes. This approach allows the model to effectively differentiate among multiple labels in the dataset.

TABLE III
SVM PARAMETERS USED IN THIS STUDY

Parameter	Value	Description
Kernel	Linear	Defines the type of hyperplane used to separate the data. Linear kernel assumes data is linearly separable in the embedding space.
C	1.0	Regularization parameter that controls the trade-off between maximizing the margin and minimizing classification error (default in scikit-learn)
Decision Function Shape	one-vs-one	Strategy used to handle multi-class classification by constructing one classifier per class pair.

In addition, other hyperparameters were kept at their default values in scikit-learn. For example, gamma is not applicable in the case of a linear kernel since it is only relevant for non-linear kernels such as RBF or polynomial. The parameters shrinking=True and max_iter=-1 (unlimited iterations) were also retained as defaults. This configuration was chosen because, in linear SVM, the kernel type and the C parameter are the most critical factors that influence performance, while other parameters have minimal effect on the classification results.

For comparison purposes, a baseline classifier was also implemented using k-Nearest Neighbor (k-NN) directly on the ResNetV1 embeddings. The choice of k-NN as a baseline is motivated by its frequent use in face recognition tasks due to its simplicity and non-parametric nature [21]. Unlike SVM, k-NN does not construct an explicit decision boundary; instead, it assigns labels based on the majority class of the nearest neighbors in the embedding space. In this study, k=3 and Euclidean distance were selected, as they are commonly used settings for embedding-based recognition. The results of this baseline experiment are reported in the Results and Discussion sections to highlight the performance gap between the proposed SVM approach and the simpler k-NN classifier.

E. Model Evaluation

512 embedding vectors applied to the SVM model to differentiate each label. After the learning phase, the datasets are applied to evaluate the SVM model, allowing it to compare the recognized face with the actual labels to determine whether they matched.

The performance of the classifier was evaluated using standard metrics, for example Precision, Recall, F1-Score. Precision measures how many positive prediction that actually true, calculated as:

$$Precision = \frac{TP}{TP + FP}$$

where TP (True Positive) represents correctly recognized faces according to the label and FP (False Positive) represents faces that are recognized incorrectly because the faces belong to another label [16].

Recall measures the proportion of actual positive samples that correctly identified by the classification model, and it is defined as:

$$Recall = \frac{TP}{TP + FN}$$

FN (False Negative) represents the faces that should have been recognized but the classifier missed the face [16].

F1-Score was generated from Precision and Recall to give balance between the two metrics [16], and it is calculated as:

$$F1 = 2 \cdot \frac{Precision\ x\ Recall}{Precision\ +\ Recall}$$

Accuracy measures overall proportion of correctly classified samples from all samples and the formula is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP and TN represent correctly classified positive and negative samples, so FP and FN represent misclassification from classification models.

Since this study deals with a multi-class dataset (191 class), the standard metrics was calculated per class, allowing it for detailed assessment and how the model can recognize each individual, highlighting strength and weakness of the model to recognize specific classes.

III. RESULTS AND DISCUSSION

A. MTCNN Face Detection Results

In this study MTCNN was applied. Out of a total 856 images, MTCNN successfully detected 832 images, achieving detection accuracy at about 97,1%.

Figure 6. Unidentified Faces

As shown in Figure 6, several images were either excessively dark or overly bright, which hindered successful face detection. This finding highlights a key limitation of

MTCNN, namely its reduced performance under excessive noise or extreme lighting conditions.

Figure 7. Identified Faces Under Extreme Conditions

From the figure 7 it can be seen that MTCNN is reliable, it can detect faces as long as the model can see the structure of the face on each photo, making MTCNN a robust model too for face detection under challenging conditions without excessive noise and lighting.

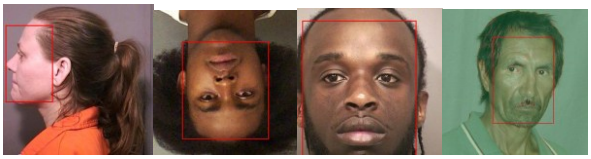

Figure 8. Identified Faces with various angle

Figure 8 proved that the MTCNN model can detect faces across various angles tested in this study, as long as the structure of the faces is visible, the MTCNN model can detect and mark the face position to be used on feature extraction.

B. Feature Extraction ResNetV1

Feature Extraction using ResNetV1 pretrained on the CASIA-WebFace dataset. Each detected and aligned face on every image from the preprocessing stage was passed to obtain the feature representation resulting in a 512-dimensional embedding vector.

For visualization purposes, only 4 labels were used, such as S001, S006, S008, S013.

TABLE IV
EMBEDDING RESULT FROM FEATURE EXTRACTION

Label	Shape
S001	(20,512)
S006	(12,512)
S008	(17,512)
S013	(13,512)

Table IV showed us the shape of embedding results based on each label. The first number represents how many images get embedded in the feature extraction process, and the last number represents the length of the embedding vector from ResNetV1.

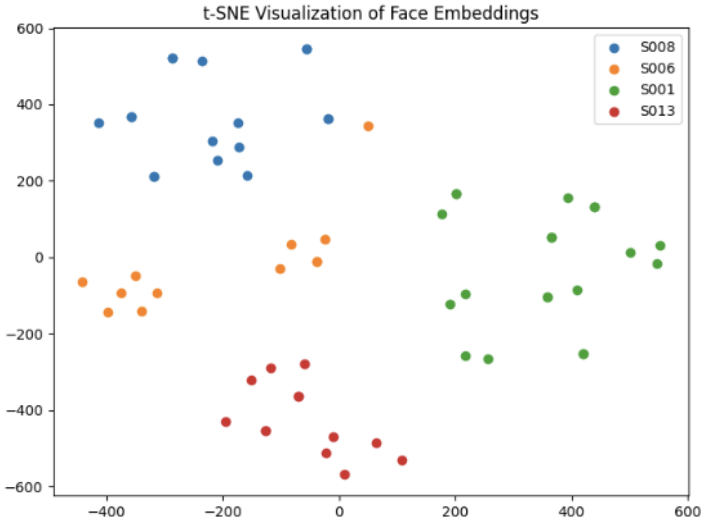

Figure 9. Visualization Result on 4 different images labels.

Figure 9 shows the t-SNE visualization of the embedding result generated by the ResNetV1 model. Each dot represents a single image embedding, while color indicates different subjects across the embedding. From the figure it can be observed that embeddings from the same subject (e.g., S001, S006, S008, S013) form distinct clusters separated from one another. Separation between clusters tells that extracted embeddings are sufficiently discriminative to identify different individuals. This experiment is crucial to identify whether this model is suitable for the learning and testing recognition process.

TABLE V
CLUSTERING PERFORMANCE EVALUATION METRICS

Metrics	Value	Interpretation
Silhouette Score	0.578	The closer to 1, the better cluster separation [17].
Davies-Bouldin Index	0.566	The smaller, the better cluster quality [18].
Calinski-Harabasz Index	111.158	The larger, the better the cluster quality [19].

Table V presents the clustering performance evaluation metrics obtained from 4 labels and t-SNE visualization of face embeddings. Silhouette Score was 0.578, it indicates that data are well clustered in between classes. Davies-Bouldin Index is 0.566, reflecting a low level of intra-cluster similarity relative to inter-cluster separation. It means a good clustering structure. The Calinski-Harabasz index was 111.158, which implies that clusters are compact and separated.

C. Single Image Recognition

This step involves testing the model with a single input image. The image is fed into the model, which then retrieves visually similar images and predicts the corresponding label that best matches the input. The model's confidence is reflected in how well the predicted label aligns with the visual features extracted from the input image.

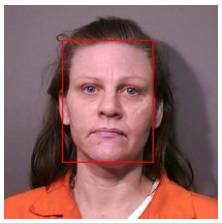

Figure 10. Single Testing Image to Test SVM.

Figure 10 shows an image that was fed into the SVM model to evaluate its performance. The test image was processed by the model, which then generated a list of labels that appeared visually similar to the input image.

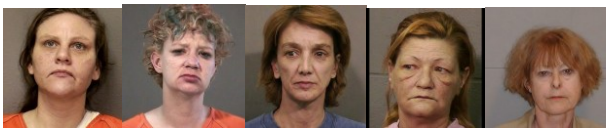

Figure 11. Images with highest similarity.

Figure 11 presents 5 images that represent each label and has the highest similarity with the testing image. The highest similarity images (From left to right: S001, S065, S134, S190, S263) is because the input image has the longest distance to S001 which is visualized by the visualisation below.

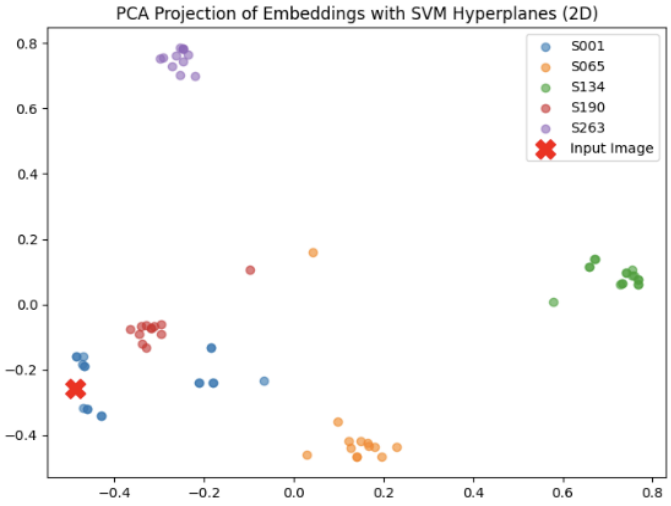

Figure 12. Hyperplane Visualization

Figure 12 presents the reason SVM recognized the input image as S001 because the testing image has the longest distance to S001, which means the model is very confident to predict the image as S001.

D. Classification using Support Vector Machine

After obtaining 512-dimensional embedding vectors. from ResNetV1, the next step is classification. SVM was applied as the classifier to differentiate between labels on the embedding classification. Each embedding that generated by ResNetV1 was assigned to SVM so the SVM can be trained with the embedding results, the objective of this process is to make SVM distinguish different subjects to make robust models for face recognition. To evaluate this classification performance, applied train-test data was 69% data for training and 31% data for testing. The model evaluated with evaluation metrics such as accuracy, precision, recall, and F1-Score.

TABLE VI
10 BEST CLASSIFICATION PERFORMANCE

Labels	Precision	Recall	F1-Score	Support
S280	1	1	1	6
S388	1	1	1	9
S392	1	1	1	5
S396	0.89	1	0.94	8
S404	1	1	1	7
S409	1	1	1	6
S414	0.78	1	0.88	7
S428	1	1	1	6
S425	1	1	1	5
S411	1	0.80	0.89	5

From table VI it can be observed 10 top performances of the experiment have consistently high values of precision, recall, f1-score. This data shows that SVM classifier combined with ResNetV1 are highly effective at recognizing several subjects.

Figure 13. Examples of correctly recognized images (From Left to Right: S388, S396, S404).

Figure 1 shows that images with clearly visible faces can be recognized perfectly. Indicating that the model can identify such samples without difficulty.

TABLE VII
10 WORST CLASSIFICATION PERFORMANCE

Labels	Precision	Recall	F1-Score	Support
S059	0	0	0	1
S120	0	0	0	1
S163	0	0	0	1
S259	0	0	0	1
S263	0	0	0	1
S285	0	0	0	1
S300	0	0	0	1
S307	0	0	0	1
S321	0	0	0	3
S336	0	0	0	1

From table VII it can be seen that the 10 worst classification performance happened in several labels. All precision, recall, F1-Score appeared to be zero. This results

indicates that the model failed to recognize the face belonging to these labels.

Figure 14. Examples of wrong recognized images (From Left to Right: S059, S120, S259).

Figure 14 shows that the model suffers from classifying images with non-frontal angles. The model can not recognize matched labels although the face alone was detected. Dark lighting affects the recognition performance also, the model can not recognize the face when the lighting is too dark.

In addition, from 336 images the classification report also provides macro average and weighted averages scores. Macro average values were 0.89 for Precision, 0.92 for Recall, and 0.89 for F1-Score. Weighted averages achieved 0.91 for precision, 0.93 for recall, and 0.91 for F1-score. These averages reinforce the reliability of the SVM classifier with ResNetV1 embeddings in handling face recognition tasks.

TABLE VIII
FAILED RECOGNIZED IMAGES

Actual Label	Predicted Label	Possible Cause
S059	S407	Non-Frontal Pose
S120	S396	Similar Face Features
S163	S214	Similar Face Features
S180	S417	Low-Lighting
S263	S174	Non-Frontal Pose

Table VIII presents five representative examples of misclassified images out of a total of 24 errors from 336 test samples. Overall, the model correctly recognized 312 images, achieving an accuracy of approximately 93%. These representative cases illustrate that while the model is generally robust, challenging conditions such as non-frontal poses, low-light environments, and high facial similarity between individuals can still lead to misclassification.

Figure 15. Example of misclassified image: subject S059 predicted as S407 due to non-frontal pose.

Figure 15 shows a representative misclassification case where subject S059 was incorrectly predicted as S407. The error occurred primarily due to the non-frontal pose, which reduced the discriminative power of the extracted features.

TABLE IX
COMPARISON OF CLASSIFICATION PERFORMANCE BETWEEN SVM AND KNN

Classifier	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
SVM	92.9%	0.89	0.92	0.89
KNN (K=3)	63.9%	0.52	0.59	0.53

From Table IX, it can be observed that the SVM classifier outperforms the baseline k-NN classifier by a significant margin. The SVM achieved an accuracy of 92.9%, with macro precision, recall, and F1-score values of 0.89, 0.92, and 0.89, respectively. In contrast, the k-NN classifier (k=3) only reached an accuracy of 63.9%, with substantially lower macro precision (0.52), recall (0.59), and F1-score (0.53). These results indicate that the proposed SVM approach is more robust and effective in handling high-dimensional embeddings generated by ResNetV1 compared to the simpler k-NN method.

E. Discussion

The experimental results demonstrate the effectiveness of the proposed face recognition pipeline, which combines MTCNN for detect face position, ResNetV1 for feature extraction, and SVM as the classifier. The face detection stage achieved an accuracy of 97.1%, successfully identifying 832 out of 856 images, as MTCNN struggled to detect faces under such conditions. Following detection, the extracted 512-dimensional embeddings using ResNetV1 proved to be highly discriminative, as visualized through t-SNE, where distinct clusters were formed for each subject. Clustering evaluation metrics further supported this, with a Silhouette Score of 0.578, a Davies-Bouldin Index of 0.566, and a Calinski-Harabasz Index of 111.158, indicating well-separated and compact clusters.

In the classification stage, the SVM model exhibited strong performance, particularly for several labels that achieved perfect precision, recall, and F1-scores, demonstrating the capability of the ResNetV1 embeddings to differentiate subjects effectively. However, some labels performed poorly, with precision, recall, and F1-score values dropping to zero. These errors mainly occurred in cases involving non-frontal face angles, poor illumination, or high similarity between subjects. Overall, the classification achieved a 93% accuracy, with macro average scores of 0.89 for precision, 0.92 for recall, and 0.89 for F1-score, while weighted averages reached 0.91, 0.93, and 0.91 respectively. These averages indicate that there are some misclassifications, the model is generally robust and reliable across diverse subjects.

Table X presents a comparison between the proposed method and several existing face recognition approaches. Haar Cascade combined with CNN achieved an accuracy of 98.94%, but its performance is limited to frontal faces and struggles when the facial pose changes. Similarly, the Viola-Jones method performs adequately on public datasets but shows a significant drop in accuracy under poor lighting and non-frontal conditions.

TABLE X
COMPARISON WITH EXISTING FACE RECOGNITION METHODS

Method	Dataset	Accuracy	Notes
Haar Cascade + CNN [1]	0 datasets	98.94%	Only works on frontal faces
Viola-Jones [3]	Public dataset	Lower under poor lighting	Struggles with non-frontal
FaceNet [11]	VGGFace2	~99%	Requires large dataset
Proposed (MTCNN + ResNetV1 + SVM)	856 images, 191 classes	93%	Works under limited dataset

FaceNet, trained on the large-scale VGGFace2 dataset, reported accuracy close to 99%, although it requires a very large dataset for optimal performance. In contrast, the proposed system that integrates MTCNN, ResNetV1, and SVM achieved 93% accuracy using only 856 images across 191 classes. Although its accuracy is slightly lower compared to large-scale deep learning models, the proposed method demonstrates robustness under limited data conditions, making it more practical for small to medium-sized datasets.

In addition to the comparison with existing methods, a baseline experiment using the k-Nearest Neighbor (k-NN) classifier was also conducted directly on the ResNetV1 embeddings. With k=3 and Euclidean distance, k-NN achieved an accuracy of only 63.9%, with macro averages of 0.52 for precision, 0.59 for recall, and 0.53 for F1-score. These results are significantly lower than those of the proposed SVM-based approach, confirming that while k-NN provides a simple non-parametric alternative, it is not sufficiently robust under the highly imbalanced and limited dataset used in this study. This performance gap underscores the importance of employing a more discriminative classifier such as SVM, which can better handle high-dimensional embeddings and achieve consistent recognition accuracy across diverse subjects.

IV. CONCLUSION

This study successfully implemented a robust face recognition pipeline by combining MTCNN for face detection, ResNetV1 for feature extraction, and SVM for classification. The dataset used consisted of 856 images across 191 classes, divided into 520 training and 336 testing samples. Experimental results demonstrate that MTCNN accurately detected 832 faces under diverse conditions, achieving a detection accuracy of 97.1%, although challenges remain in cases with extreme lighting. The extracted 512-dimensional embeddings from ResNetV1 proved to be highly discriminative, forming well-separated clusters and achieving strong clustering evaluation metrics, including a Silhouette Score of 0.578, a Davies-Bouldin Index of 0.566, and a Calinski-Harabasz Index of 111.158.

At the classification stage, the SVM achieved an overall accuracy of 93%, with macro average scores of 0.89 for Precision, 0.92 for Recall, and 0.89 for F1-Score, while weighted averages reached 0.91, 0.93, and 0.91 respectively. Despite some misclassifications in cases involving non-frontal angles, dark lighting, and visually similar subjects, the proposed system demonstrates strong reliability. Therefore, the integration of MTCNN, ResNetV1, and SVM provides an effective solution for face recognition tasks and shows great potential for real-world applications such as security systems, identity verification, and automated attendance systems.

This work contributes to the understanding of face recognition in resource-constrained settings, where datasets are limited in size and subject images are scarce. Such conditions are common in real-world deployments, particularly in small institutions or organizations, making this study practically relevant.

Unlike many face recognition approaches that rely on large-scale datasets and end-to-end deep learning models, this study highlights the effectiveness of the pipeline (MTCNN + ResNetV1 + SVM) in handling small and imbalanced datasets while remaining robust against common challenges such as pose variation and lighting changes. This novelty underscores the practicality and efficiency of the proposed system, particularly for real-world deployments where data resources are limited.

REFERENCES

[1] B. Hartika and D. Ahmad, "Face Recognition Menggunakan Algoritma Haar Cascade Classifier dan Convolutional Neural Network," *Journal of Mathematics UNP*, vol. 2, no. 1, pp. 1–7, 2022. [Online]. Available: https://ejournal.unp.ac.id/students/index.php/mat/article/view/11954

[2] O. A. Naser, S. Mumtazah, K. Samsudin, M. Hanafi, S. M. Binti, and N. Z. Zamri, "Comparative Analysis of MTCNN and Haar Cascades for Face Detection in Images with Variation in Yaw Poses and Facial Occlusions," *Journal of Communications Software and Systems*, vol. 21, no. 1, pp. 109–119, Mar. 2025, doi: 10.24138/jcomss-2024-0084.

[3] D. M. Abdulhussien and L. J. Saud, "Evaluation Study of Face Detection by Viola-Jones Algorithm," *International journal of health sciences*, pp. 4174–4182, Sep. 2022, doi: https://doi.org/10.53730/ijhs.v6ns8.13127.

[4] F. R. Chandra, A. Nur, and R. Hidayat, "Analysis of the use of MTCNN and landmark technology to improve the accuracy of facial recognition on official documents," *Journal of Applied Informatics and Computing*, vol. 9, no. 2, pp. 112–120, 2025. [Online]. Available: https://jurnal.polibatam.ac.id/index.php/JAIC/article/view/8814

[5] Z. Li *et al.*, "A classification method for multi-class skin damage images combining quantum computing and Inception-ResNet-V1," *Frontiers in Physics*, vol. 10, Nov. 2022, doi: https://doi.org/10.3389/fphy.2022.1046314.

[6] S. Almabdy and L. Elrefaei, "Deep Convolutional Neural Network-Based Approaches for Face Recognition," *Applied Sciences*, vol. 9, no. 20, p. 4397, Oct. 2019, doi: https://doi.org/10.3390/app9204397.

[7] R. E. Saragih and Q. H. To, "A Survey of Face Recognition based on Convolutional Neural Network," *Indonesian Journal of Information Systems*, vol. 4, no. 2, Feb. 2022, doi: https://doi.org/10.24002/ijis.v4i2.5439.

[8] P. P. Raj, "An Evaluation of MTCNN in Face Recognition Algorithms for Effective Detection in Masked Scenarios in Real Time Video Surveillance," *African Journal of Biomedical Research*, pp. 12252–12261, Dec. 2024, doi: https://doi.org/10.53555/ajbr.v27i4s.6154.

[9] M. Yuan, Seyed Yahya Nikouei, Alem Fitwi, Y. Chen, and Y. Dong, "Minor Privacy Protection Through Real-time Video Processing at the Edge," *arXiv (Cornell University)*, Aug. 2020, doi: https://doi.org/10.1109/icccn49398.2020.9209632.

[10] "A. I. Awodeyi, O. A. Ibok, I. Omokaro, J. U. Ekwemuka, and M. O. Ighofiomoni, "Effective preprocessing techniques for improved facial recognition under variable conditions," *Franklin Open*, vol. 10, p. 100225, Jan. 2025, doi: https://doi.org/10.1016/j.fraope.2025.100225.

[11] A. Jan, S. Abid, M. F. Khan, A. Hussain, and A. A. Khuhro, "Evaluation of Pre-Trained CNN Models for Face Recognition," *Sensors*, vol. 23, no. 6, p. 2901, 2023. [Online]. Available: https://doi.org/10.3390/s23062901

[12] I. C. Duta, L. Liu, F. Zhu, and L. Shao, "Improved Residual Networks for Image and Video Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4135–4147, Dec. 2021, doi: https://doi.org/10.1109/TPAMI.2021.3055457

[13] Y. Yang *et al.*, "A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions," *Computers in Biology and Medicine*, vol. 139, p. 104887, Dec. 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104887.

[14] P. Hofer, M. Roland, P. Schwarz, and R. Mayrhofer, "Shrinking embeddings, not accuracy: Performance-preserving reduction of facial embeddings for complex face verification computations," Johannes Kepler Univ. Linz, Austria, Tech. Rep., 2023. [Online]. Available: https://www.researchgate.net/publication/384278329

[15] H. Zhang, "Real-time face recognition method based on MTCNN-Inception-ResNet-v2-SVM model," *Applied and Computational Engineering*, vol. 45, no. 1, pp. 179–189, Mar. 2024, doi: https://doi.org/10.54254/2755-2721/45/20241677.

[16] J. Opitz, "A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 820–836, Jan. 2024, doi: https://doi.org/10.1162/tacl_a_00675.

[17] G. Vardakas, I. Papakostas, and A. Likas, "Deep clustering using the soft silhouette score: Towards compact and well-separated clusters," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[18] Yuli Asriningtias and Joko Aryanto, "K-Means Algorithm with Davies Bouldin Criteria for Clustering Provinces in Indonesia Based on Number of Events and Impacts of Natural Disasters," *International Journal of Engineering Technology and Natural Sciences*, vol. 4, no. 1, pp. 75–80, Jul. 2022, doi: https://doi.org/10.46923/ijets.v4i1.147.

[19] Z. Syahputri, Sutarman, dan M. A. P. Siregar, "Determining The Optimal Number of K-Means Clusters Using The Calinski Harabasz Index and Krzanowski and Lai Index Methods for Grouping Flood Prone Areas In North Sumatra," *SinkrOn*, vol. 8, no. 1, pp. 571–580, Jan. 2024.

[20] H. Tariq, M. Majeed, and M. Ahmad, "Optimizing SVM Performance through Combinatorial Hyperparameter Tuning and Model Selection," Univ. of Agriculture Faisalabad, Pakistan, 2025. [Online]. Available: https://www.researchgate.net/publication/393104826_Optimizing_SVM_Performance_through_Combinatorial_Hyperparameter_Tuning_and_Model_Selection

[21] N. R. Feta, "Comparison of KNN and SVM Algorithms in Facial Image Recognition Using Haar-Wavelet Feature Extraction," *Information Systems and Technology, Indonesia Cyber University*, 2023. [Online]. Available: https://www.researchgate.net/publication/371876569_Comparison_of_KNN_and_SVM_Algorithms_in_Facial_Image_Recognition_Using_Haar_Wavelet_Feature_Extraction. Accessed: Sep. 8, 2025.