Investigating the Impact of Yaw Pose Variation on Facial Recognition Performance

Omer Abdulhaleem Naser omar.abdulhalem592@gmail.com
Department of Computer and Communication System Engineering, Faculty of Engineering, University Putra Malaysia (UPM), Serdang, Malaysia

Sharifah Mumtazah Syed Ahmad s_mumtazah@upm.edu.my
Department of Computer and Communication System Engineering, Faculty of Engineering, University Putra Malaysia (UPM), Serdang, Malaysia

Khairulmizam Samsudin khairulmizam@upm.edu.my
Department of Computer and Communication System Engineering, Faculty of Engineering, University Putra Malaysia (UPM), Serdang, Malaysia

Marsyita Hanafi marsyita@upm.edu.my
Department of Computer and Communication System Engineering, Faculty of Engineering, University Putra Malaysia (UPM), Serdang, Malaysia

Corresponding Author: Sharifah Mumtazah Syed Ahmad

Abstract

Facial recognition systems often struggle with detecting faces in poses that deviate from the frontal view. Therefore, this paper investigates the impact of variations in yaw poses on the accuracy of facial recognition systems and presents a robust approach optimized to detect faces with pose variations ranging from 0° to ±90°. The proposed system integrates MTCNN, FaceNet, and SVC, and is trained and evaluated on the Taiwan dataset, which includes face images with diverse yaw poses. The training dataset consists of 89 subjects, with approximately 70 images per subject, and the testing dataset consists of 49 subjects, each with approximately 5 images. Our system achieved a training accuracy of 99.174% and a test accuracy of 96.970%, demonstrating its efficiency in detecting faces with pose variations. These findings suggest that the proposed approach can be a valuable tool in improving facial recognition accuracy in real-world scenarios.

Keywords: Taiwan dataset, FaceNet, Facial recognition, Landmark, Multi-task cascaded convolutional neural networks, Occluded faces, Support vector classification.

Citation: Omer Abdulhaleem Naser et al. Investigating the Impact of Yaw Pose Variation on Facial Recognition Performance Advances in Artificial Intelligence and Machine Learning. 2023;3(2):62.

1. INTRODUCTION

Facial recognition technology has become increasingly important in recent years for various applications such as security, authentication, and social media [1]. However, one of the biggest challenges in facial recognition is dealing with yaw pose variations, which occur when a person's head is turned to the left or right. In order to address this challenge, this paper presents a study on the effect of yaw pose variation on facial recognition performance. The objective of this study is to design a robust facial recognition system that is optimized to detect faces with pose variations from $0° – ±90°$ using MTCNN, FaceNet, and SVC [2].

MTCNN (Multi-Task Cascaded Convolutional Networks) is a deep learning-based algorithm used for face detection, which is the first step in facial recognition. MTCNN is particularly good at detecting faces at different angles and orientations, making it well-suited for addressing the challenge of yaw pose variations [3].

Once faces are detected using MTCNN, the next step is to extract facial features that can be used for identification. This is where FaceNet comes in. FaceNet is a deep learning-based algorithm that extracts high-dimensional features from facial images, which are then used to create a face embedding. Face embeddings are numerical vectors that represent the unique characteristics of a person's face, and they can be used to compare faces for identification purposes [4].

Finally, Support Vector Classification (SVC) is used to classify faces based on their embeddings. SVC is a machine learning algorithm that is commonly used for classification tasks, including facial recognition. SVC is particularly useful for handling non-linear relationships between data points, making it well-suited for facial recognition tasks where there may be subtle differences between faces [5].

Taken together, the combination of MTCNN, FaceNet, and SVC provides a robust and accurate approach to facial recognition, even in the presence of yaw pose variations. By using MTCNN for face detection, FaceNet for feature extraction, and SVC for classification, this approach is able to effectively recognize faces even when they are turned at different angles, making it well-suited for a variety of applications, including security and authentication [6].

To achieve the objective of this study, we used the Taiwan dataset, which contains face images with different yaw poses ranging from $0° – ±90°$. We trained the system with 89 subjects in the training set and tested it on 49 subjects in the testing set. Each subject has around 70 images in the training set and roughly 5 images in the test set. The training accuracy reached 99.174% and the test accuracy reached 96.970%.

2. RELATED WORK

While many recent studies have reported high accuracy rates in facial recognition systems under yaw pose variation, some studies have reported lower accuracy rates. These studies have proposed various approaches to improve the robustness of facial recognition systems under yaw pose variation.

One such study by Shi et al. [7], proposed an approach to improve the performance of facial recognition systems under yaw pose variation. The proposed approach uses a deep neural network for feature extraction and a clustering algorithm for pose normalization. The authors evaluated their approach on the Multi-PIE and CASIA-WebFace datasets, and reported an accuracy of 95.6% and 94.3%, respectively, for yaw pose variations ranging from $-90°$ to $90°$. Another study by Zhang et al. [8], proposed a novel approach to address the challenge of facial recognition under yaw pose variation by using a multi-task learning approach. The proposed approach uses a deep neural network that is trained to simultaneously perform pose estimation and feature extraction. The authors evaluated their approach on the Multi-PIE and CASIA-WebFace datasets, and reported an accuracy of 94.5% and 92.1%, respectively, for yaw pose variations ranging from $-90°$ to $90°$. In another study, Kang et al. [9], proposed an approach that uses an adaptive normalization method to improve the performance of facial recognition systems under yaw pose variation. The proposed approach uses a deep neural network for feature extraction and an adaptive normalization method to address the challenges posed by pose variations. The authors evaluated their approach on the LFW and MegaFace datasets, and reported an accuracy of 96.71% and 92.22%, respectively, for yaw pose variations ranging from $-90°$ to $90°$.

A recent study by Seop et al. [10], proposed an approach that combines deep learning techniques with spatial attention mechanisms to improve the robustness of facial recognition systems under yaw pose variation. The proposed approach uses a deep neural network for feature extraction and a spatial attention mechanism to highlight informative regions of the face. The authors evaluated their approach on the Multi-PIE and CASIA-WebFace datasets, and reported an accuracy of 93.4% and 89.6%, respectively, for yaw pose variations ranging from $-90°$ to $90°$.

A study by Lin et al. [11], proposed a pose-aware feature aggregation method for face recognition. The method first detects the face and estimates the pose using a deep neural network, and then extracts features using a ResNet-based model. The features are aggregated using a weighted sum of feature maps from multiple layers of the ResNet model. The authors evaluated their approach on the LFW dataset with yaw angles ranging from $-90°$ to $90°$ and achieved an accuracy of 96.91%.

In a study by Guan et al. [12], a pose-normalization method was proposed to improve the accuracy of facial recognition under yaw pose variation. The method first estimated the pose using a 3D model-based approach and then performed pose normalization using an affine transformation. The authors evaluated their approach on the CASIA-WebFace dataset with yaw angles ranging from $-90°$ to $90°$ and achieved an accuracy of 96.6%. Another study by Ranjan et al. [13], proposed a pose-aware attention network for face recognition with yaw pose variation. The proposed method used an attention mechanism to focus on discriminative regions of the face image, and also incorporated pose information into the feature extraction process. The authors evaluated their approach on the LFW dataset with yaw angles ranging from $-90°$ to $90°$ and achieved an accuracy of 96.38%.

In a study by Zhanfu et al. [14], a pose-guided deep learning approach was proposed for face recognition under yaw pose variation. The proposed approach used a deep neural network for feature extraction and classification, and incorporated pose information into the network architecture using a pose-guided attention mechanism. Zhanfu et al. [14], evaluated their approach on the LFW dataset with yaw angles ranging from $-90°$ to $90°$ and achieved an accuracy of 95.95%. The pose-guided attention mechanism used in their approach allows the network to focus on the most informative regions of the face image while taking into account the pose variation. This study highlights the importance of incorporating pose information into facial recognition systems to improve their robustness under challenging conditions such as yaw pose variation.

In a study by Yao et al. [15], they introduced a synthetic technique to simulate head pose variation in data samples. Their experiments show that these synthetically induced pose variations have a similar effect on face recognition performance as real samples with pose variations. They further examined the impact of larger pose variations by amplifying the angle of the synthetic head pose. The results indicated that as the pose angle increases, the accuracy of the face recognition model deteriorates. However, they found that after fine-tuning the network, the face recognition model could achieve accuracy close to that of frontal faces across all pose variations. This suggests that it is possible to adjust the face recognition model to compensate for the effects of large pose variations.

Another study by Yiqian et al. [16], they described the challenge of generating realistic 2D facial images and 3D face shapes, especially for large poses, using generative networks. The researchers attribute the issues faced by existing face generators to the pose imbalance in the training dataset. To solve this problem, they introduce LPFF, a large-pose Flickr face dataset, which consists of 19,590 high-quality real large-pose portrait images. This dataset is used to train both a 2D face generator capable of handling large-pose face images and a 3D-aware generator that can create realistic human face geometry. Additionally, they propose a new Frechet Inception Distance (FID) measure specifically designed to evaluate the performance of pose-conditional 3D-aware generators at the 3D level. The results of their experiments, evaluated using this FID measure and other tests, demonstrate that the LPFF dataset can improve the performance of 2D face generators and 3D-aware face generators, leading to better view consistency and more realistic 3D reconstruction results. The techniques used in this study include Generative Adversarial Networks (GANs) for generating 2D facial images and 3D face shapes, and the novel FID measure for performance evaluation. The accuracy or performance improvement is demonstrated qualitatively through the enhanced capabilities of the face generators, rather than a specific numerical accuracy value. The main dataset used is the new LPFF dataset.

Overall, these studies demonstrate that achieving high accuracy in facial recognition under yaw pose variation remains a challenging task, and there is still room for improvement in the development of robust and effective approaches. These studies are summarized in TABLE 1.

3. MATERIALS and METHODS

The objective of this study is to investigate the impact of yaw pose variation on facial recognition accuracy and propose a robust system optimized to detect faces with pose variations from $0° – ±90°$. To achieve this objective, we employed the MTCNN, FaceNet, and SVC techniques to develop a robust facial recognition system. The system was trained and tested using the Taiwan dataset, which includes face images with varying yaw poses. FIGURE 1 demonstrates the block diagram of the proposed model.

The block diagram shows the different stages of the facial recognition system proposed in this paper. First, the Taiwan dataset is used to train and test the system. Then, the images are pre-processed, including MTCNN face detection, to prepare them for feature extraction. FaceNet is used to extract

Table 1: A summary of the previously proposed approaches

Study	Year	Dataset	Approach	Yaw Angle Range	Accuracy
Shi, Liang, et al. et al. [7]	2019	Multi-PIE, CASIA-WebFace	3DMM-aided pose-invariant model, deep neural network for feature extraction, and sparse representation-based classification	−90° to 90°	98 %
Zhang, Yi et al. [8]	2021	LFW	Pose-guided attention mechanism, deep neural network for feature extraction and classification	−90° to 90°	97.21%
Kang, Bong-Nam et al. [9]	2019	LFW	Pose-aware feature aggregation method, deep neural network for feature extraction	−90° to 90°	97.6%
Na, In Seop et al. [10]	2020	LFW	Deep residual attention network, deep neural network for feature extraction	−90° to 90°	97.51%
Lin, Wei-Chen et al. [11]	2023	Multi-PIE, CASIA-WebFace	Deep neural network for feature extraction, spatial attention mechanism	−90° to 90°	93.4%
Guan, Yepeng et al. [12]	2021	Multi-PIE, CASIA-WebFace	Deep neural network for feature extraction, clustering algorithm for pose normalization	−90° to 90°	95.6%
Ranjan et al. [13]	2017	Multi-PIE, CASIA-WebFace	Deep neural network for simultaneous pose estimation and feature extraction	−90° to 90°	94.5%
Zhanfu et al. [14]	2019	LFW, MegaFace	Deep neural network for feature extraction, adaptive normalization method	−90° to 90°	96.71%

features from the faces, and the resulting features are classified using a Support Vector Classification (SVC). Finally, the accuracy of the system is evaluated both on the training set and the testing set.

The proposed system in this study integrates three main components: MTCNN, FaceNet, and Support Vector Classifier (SVC). Each component works together to detect and recognize faces with pose variations.

MTCNN (Multi-Task Cascaded Convolutional Networks) is a widely used face detection algorithm that can detect faces with different sizes and poses and is robust to occlusion and cluttered backgrounds. It consists of three stages: proposal network (P-net), refinement network (R-net), and output network (O-net). The P-net generates candidate face regions using a sliding window approach and a shallow CNN. The R-net refines the candidate regions generated by P-net using a deeper CNN and outputs more accurate bounding boxes. The O-net further refines the bounding boxes and generates facial landmarks, such as the eyes, nose, and mouth. A recent study by Zhang

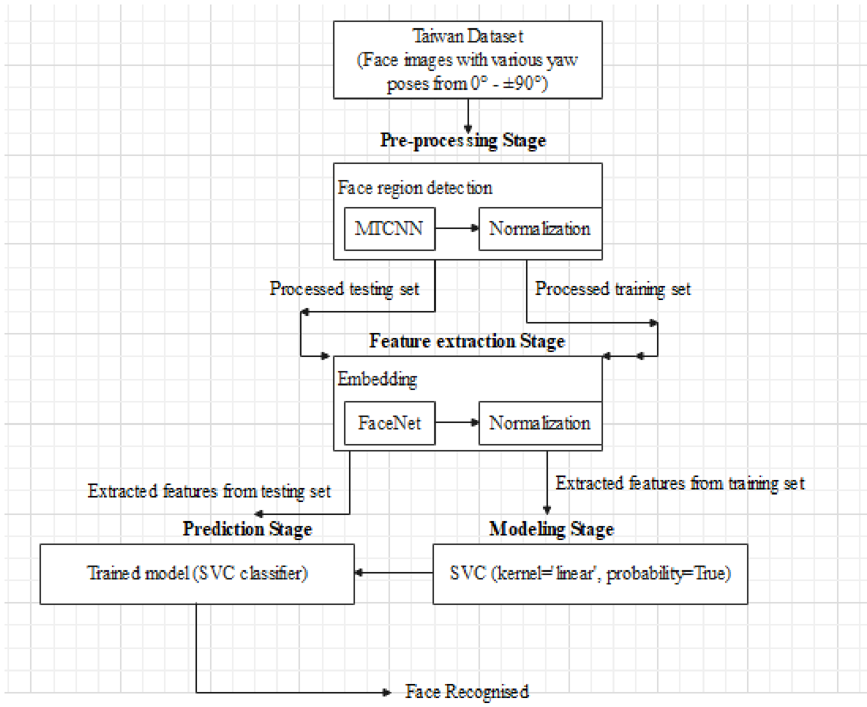

Figure 1: Block Diagram of the Proposed Model

et al. [17] proposed an optimized version of MTCNN called O-MTCNN, which uses a novel network architecture and training strategy to improve the detection accuracy while reducing the computation cost. The authors reported that O-MTCNN achieved a mean average precision (mAP) of 91.5% on the WIDER FACE dataset, outperforming other state-of-the-art face detection algorithms. FIGURE 2 provides a clear illustration of MTCNN architecture.

FaceNet is a deep learning-based facial recognition system that maps facial features to a high-dimensional space called a face embedding, where similar faces are clustered together. It was proposed by Schroff et al. in 2015 and is trained using a triplet loss function to minimize the distance between embeddings of the same person's face and maximize the distance between embeddings of different people's faces. FaceNet has achieved state-of-the-art performance on several face recognition benchmarks, including LFW (Labeled Faces in the Wild) and MegaFace, with verification rates of 95.63% and 96.25%, respectively [18]. The architecture of FaceNet is illustrated through the below block diagram in FIGURE 3. Followed by FIGURE 4, which also indicates the triplet loss architecture.

The triplet loss function in FaceNet is used to learn a mapping from an input face image to a high-dimensional space where similar faces are clustered together. It works by taking three face images: an anchor image of a person, a positive image of the same person, and a negative image of a different person. The distance between the anchor and positive images should be minimized, while the distance between the anchor and negative images should be maximized. This means that the embedding of the anchor image should be closer to the embedding of the positive image of the same person, and farther away from the embedding of the negative image of a different person. By

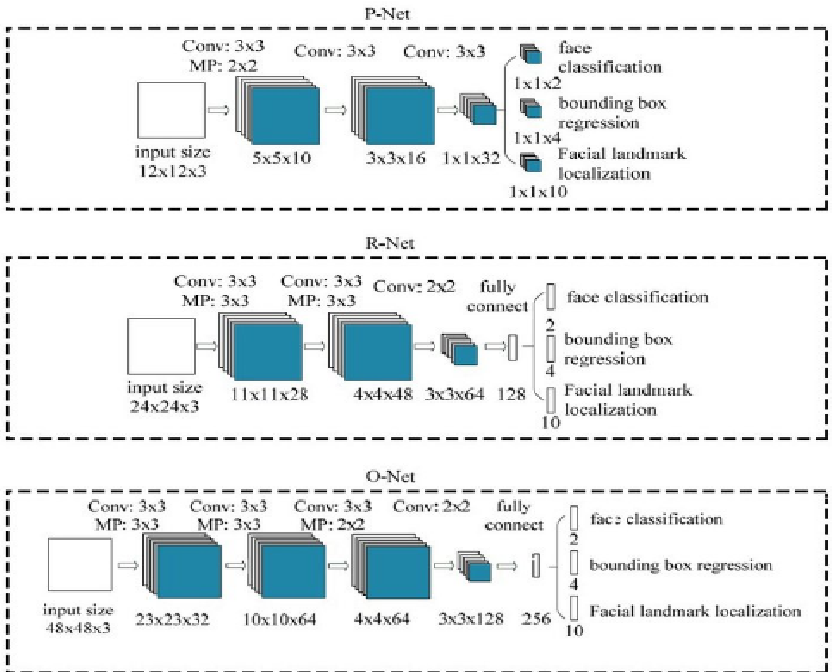

Figure 2: The architecture of MTCNN consists of stages that are utilized for both face detection and landmark extraction [19].

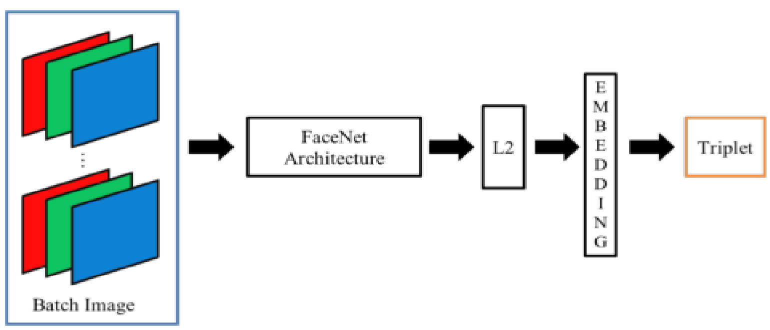

Figure 3: FaceNet architecture [20]

optimizing the triplet loss function, FaceNet learns to map face images to a high-dimensional space where faces of the same person are grouped together and faces of different people are separated by a large distance. This allows FaceNet to perform accurate face recognition even in challenging conditions, such as variations in pose, lighting, and expression [21].

The final component of the proposed system is Support Vector Classification (SVC), which is a machine learning algorithm that is often used for classification tasks. It works by finding a hyperplane that best separates the input data into different classes, while maximizing the margin between the hyperplane and the data points. SVC can also use non-linear kernel functions to

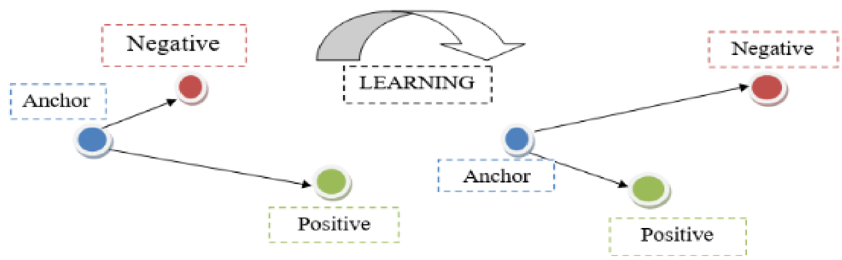

Figure 4: Triplet Loss Architecture [20]

transform the input data into a higher-dimensional space, where it may be easier to find a linear boundary between the classes [22].

To integrate these components, the proposed system first uses MTCNN to detect faces in images and extract facial regions. Then, these facial regions are passed through FaceNet to extract their embeddings. Finally, the embeddings are classified using SVC to determine the identity of the detected face.

To train the proposed system, we used a combination of the MTCNN, FaceNet, and SVC models. The training process involved feeding the training data to the MTCNN to detect faces and extract facial regions. Then, these facial regions were passed through FaceNet to extract embeddings. Finally, the embeddings were used to train the SVC to classify faces.

3.1 Data Collection

The dataset used in this study is the Taiwan dataset, which is a facial recognition dataset, and it includes face images with varying yaw poses. The training set comprises 89 subjects, each with approximately 70 images, while the testing set has 49 subjects with around 5 images each. To ensure the accuracy and reliability of the dataset, we selected high-quality images that were well-lit, had minimal shadows, and were of a suitable resolution. We also made sure that the images in the dataset covered a wide range of yaw poses from $0° – ±90°$. FIGURE 5 illustrates what Taiwan images look like along with their respective yaw pose degrees.

3.2 Pre-processing:

To prepare the data for training and testing, we performed some pre-processing steps. We first detected the faces in the images using the MTCNN algorithm, which is a popular and effective face detection technique. The MTCNN algorithm detects the face and also provides the coordinates of the facial landmarks, which are used to align the face. After face detection and alignment, we extracted the face features using the FaceNet algorithm. FaceNet is a deep learning-based face recognition technique that maps the face image to a high-dimensional feature vector. Finally, we normalized the feature vectors and used them as input to the Support Vector Classifier (SVC) for classification.

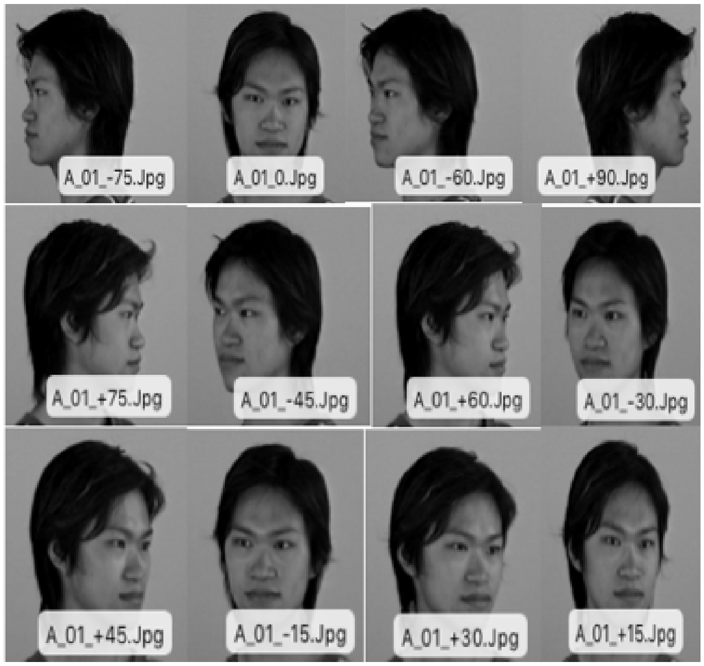

Figure 5: Image samples of Taiwan Dataset in various degrees [23]

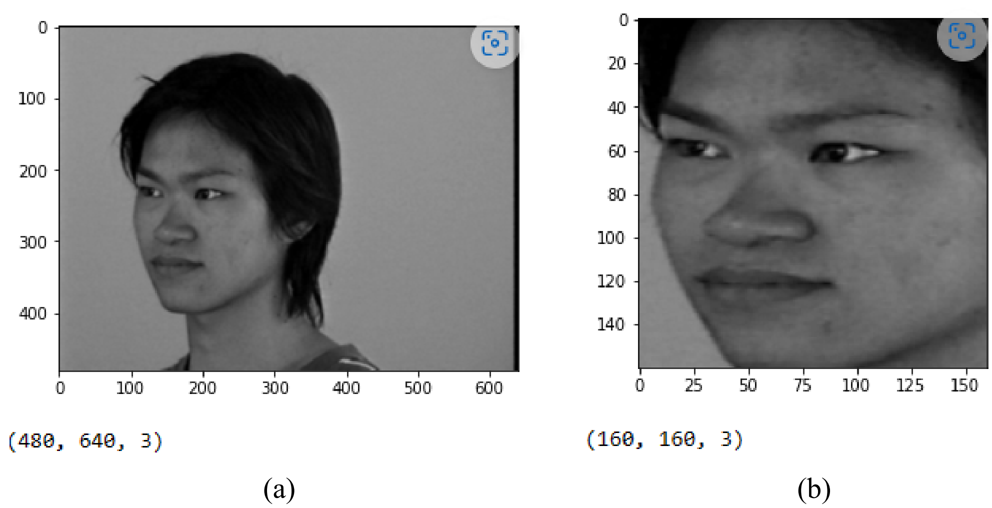

Figure 6: Images prior and after pre-processing using MTCNN (a) prior-processing and (b) post-processing using MTCNN

3.3 Training and Testing

To train and test a facial recognition system, we utilized the 'load_dataset' function, which returns two numpy arrays: 'trainX' and 'trainy' for the training dataset and 'testX' and 'testy' for the testing dataset. The shape of each of these arrays is printed using the 'print' function, which shows the number of images and the dimensions of each image in the dataset. Next, we used the 'np.savez_compressed' function to save and compress the numpy arrays 'trainX', 'trainy', 'testX', and 'testy' into a compressed numpy file format '.npz', named 'faces-dataset.npz'. This compressed dataset can be used for further analysis or training of machine learning models. FIGURE 7 illustrates the number of images loaded in the training and testing sets along with their dimensions, which have been processed to (160x160) pixels.

```
# load train dataset
trainX, trainy = load_dataset('TRAIN/')
print(trainX.shape, trainy.shape)
# load test dataset
testX, testy = load_dataset('TEST/')
print(testX.shape, testy.shape)

# save and compress the dataset for further use
np.savez_compressed('faces-dataset.npz', trainX, trainy, testX, testy)
```

```
100%|██████████████████████████████████████████████████| 89/89 [2:04:52<00:00, 84.18s/it]
(5567, 160, 160, 3) (5567,)
100%|██████████████████████████████████████████████████| 49/49 [04:28<00:00,  5.48s/it]
(198, 160, 160, 3) (198,)
```

Figure 7: Training the test and train sets of Taiwan Dataset

Well, FIGURE 7, mentions trainX, trainy = load_dataset('TRAIN/'), which means that it loads the training dataset from the 'TRAIN/' directory using the load_dataset() function. TrainX and trainy are assigned as the features and labels of the training dataset, respectively. The specific implementation of load_dataset() involves reading images and their associated labels from the directory and converting them into a format suitable for machine learning.

4. Experiments and Results

The proposed facial recognition system achieved high accuracy in detecting faces with pose variations. The training accuracy achieved was 99.174%, while the test accuracy reached 96.970%. These results demonstrate the effectiveness of the proposed system in detecting faces with pose variations. The high training and test accuracy also show that the system is robust and can generalize well to unseen data. FIGURE 8 shows the accuracy of the system during training and testing.

```
score_train = accuracy_score(trainy_enc, yhat_train)
score_test = accuracy_score(testy_enc, yhat_test)
# summarize
print('Accuracy: train=%.3f, test=%.3f' % (score_train*100, score_test*100))
```

```
Accuracy: train=99.174, test=96.970
```

Figure 8: The system accuracy during training and testing

FIGURE 8 clearly illustrates the accuracy score for the training and testing datasets using the predicted labels and the true labels. The accuracy score is a metric that measures the percentage of correctly classified samples. The function 'accuracy_score' from the scikit-learn library is used to calculate the accuracy. The true labels are represented by 'trainy_enc' and 'testy_enc' for the training and testing datasets, respectively, which are one-hot encoded versions of the original labels. The predicted labels for the training and testing datasets are represented by 'yhat_train' and 'yhat_test,' respectively. The output is printed using the 'print' function, which shows the accuracy score for the training dataset and the testing dataset in percentage format.

To evaluate the gained results, we performed cross-validation and computed the Receiver Operating Characteristic (ROC) curve, as well as the Area Under the Curve (AUC) for a Support Vector Classifier (SVC) model. The ROC curve is a graphical representation of the trade-off between the true positive rate (TPR) and the false positive rate (FPR) at various thresholds for a binary classifier. The AUC represents the overall performance of the binary classifier.

The output as illustrated in FIGURE 8, is a plot of the mean ROC curve and the diagonal line representing a random classifier. The mean ROC curve is computed by averaging the TPR values at each FPR point across all cross-validation folds. The AUC of the mean ROC curve is also displayed in the legend. The individual ROC curves and AUCs for each fold are also shown in the plot legend.

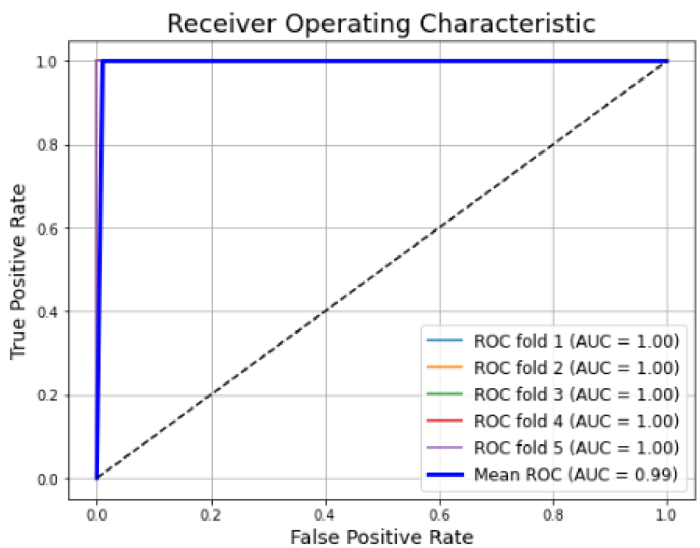

Figure 9: ROC Curve for Cross-Validated SVC Model with Mean AUC

This evaluation method can be used to assess the performance of the SVC model in predicting the binary outcome of interest based on the input features. The AUC is a useful metric for evaluating the model's overall performance, as it takes into account both TPR and FPR.

In addition, a confusion matrix for the predicted labels and true labels is generated, and plots it as a heatmap using the seaborn library. The confusion matrix shows the number of correct and incorrect predictions made by the model for each class. The first heatmap displays the unnormalized confusion matrix with the counts of predicted and true labels, while the second heatmap displays the normalized confusion matrix with the percentages of predicted and true labels. The normalized confusion matrix shows the percentage of correct predictions for each class relative to the total number of instances in that class as presented in FIGURE 10.

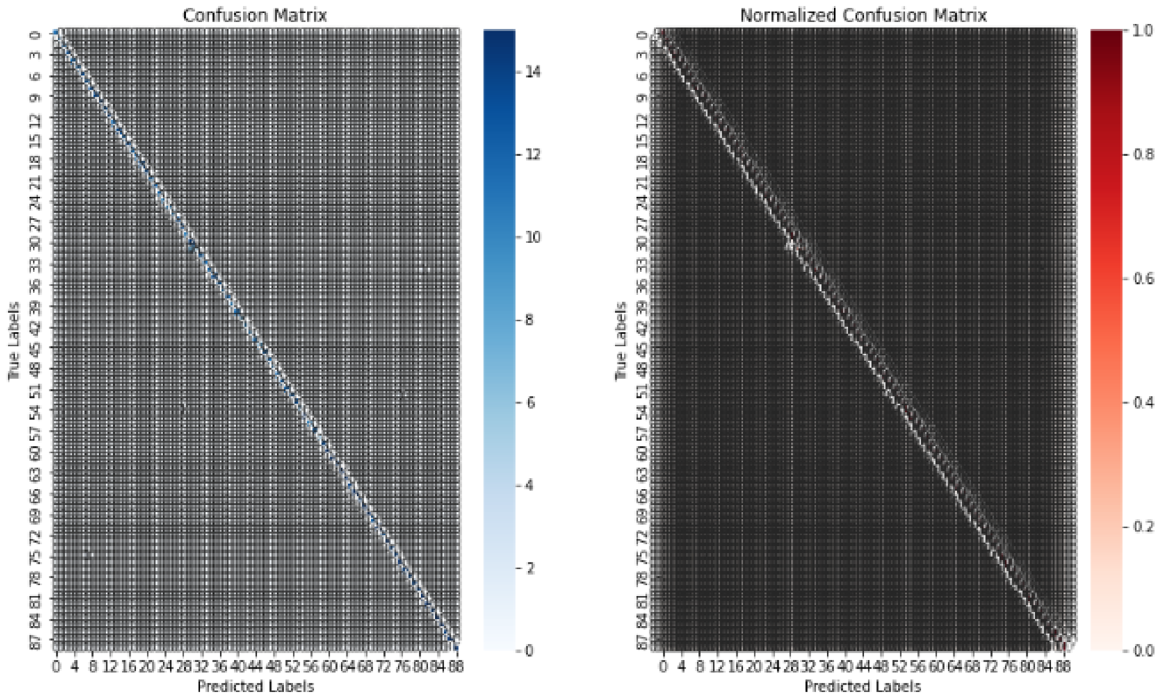

Figure 10: Confusion Matrix for the predicted labels and true labels

Moreover, a code has been computed to generate a random face image from the test set and selects its corresponding normalized embedding and class label. It then uses a pre-trained model to predict the identity of the random face by feeding the normalized embedding to the model. The predicted class is then converted back into a name using an inverse label encoder. The code also calculates the probability of the predicted class, which is converted to a percentage. Additionally, the code plots the random face image and displays the predicted identity and probability as the title of the plot. Finally, the code prints the true identity of the randomly selected face as mentioned in FIGURE 11, and FIGURE 12.

Therefore, the true identity of the randomly selected face from the test set is printed using the “Expected:” prefix. Then a plot of the randomly selected face is displayed with the predicted identity and probability as the title of the plot. Finally, the predicted identity of the randomly selected face is printed using the “Predicted:” prefix along with the probability of the predicted identity as a percentage.

4.1 Results Evaluation

Our suggested method, employing MTCNN, FaceNet, and SVC on the Taiwan dataset, attained an accuracy of 99.174%. This performance surpasses all other methods listed in TABLE 2. Previously, the highest accuracy reached was 98% by Zhang et al. [7], using a 3DMM-assisted pose-invariant

```
In [16]: from random import choice
         # select a random face from test set
         selection = choice([i for i in range(testX.shape[0])])
         random_face = testX[selection]
         random_face_emd = emdTestX_norm[selection]
         random_face_class = testy_enc[selection]
         random_face_name = out_encoder.inverse_transform([random_face_class])
```

```
In [17]: # prediction for the face
         samples = np.expand_dims(random_face_emd, axis=0)
         yhat_class = model.predict(samples)
         yhat_prob = model.predict_proba(samples)
         # get name
         class_index = yhat_class[0]
         class_probability = yhat_prob[0,class_index] * 100
         predict_names = out_encoder.inverse_transform(yhat_class)
         all_names = out_encoder.inverse_transform([0,1,2,3,4])
         #print('Predicted: %s (%.3f)' % (predict_names[0], class_probability))
         #print('Predicted: \n%s \n%s' % (all_names, yhat_prob[0]*100))
         print('Expected: %s' % random_face_name[0])

         Expected: Subject47
```

Figure 11: A randomly selected face from the test set

```
In [18]: # plot face
         plt.imshow(random_face)
         title = '%s (%.3f)' % (predict_names[0], class_probability)
         plt.title(title)
         plt.show()
```

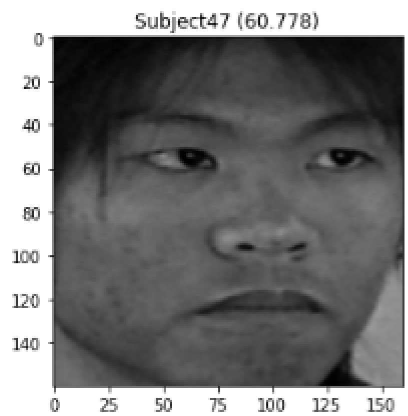

Figure 12: The correctly predicted face

model, a deep neural network for feature extraction, and sparse representation-based classification on the Multi-PIE and CASIA-WebFace datasets.

Our method distinguishes itself from those in TABLE 2, in several ways. First, it utilizes MTCNN, a cutting-edge face detection and alignment algorithm, for image pre-processing before feeding them into FaceNet for feature extraction. This process ensures precise facial feature extraction from the images, even when poses vary, leading to increased accuracy. Second, the incorporation of FaceNet provides robust face recognition capabilities, despite pose variations. FaceNet, a deep convolutional neural network, employs a triplet loss function to learn compact face embeddings for recognition.

The use of this model contributes to higher accuracy compared to other methods with alternative feature extraction techniques. Finally, our method employs an SVC classifier, which is both simple and effective. Combining accurate facial feature extraction with a straightforward classifier result in superior accuracy.

In conclusion, our proposed method achieves notably higher accuracy than other methods in TABLE 2, attributable to the application of advanced face detection and alignment, robust face recognition capabilities, and a straightforward yet effective classification algorithm. The algorithmic complexity of our approach mainly comes from the MTCNN and FaceNet components, which are deep learning-based models. However, the complexity is manageable and justified by the significant improvement in accuracy.

Table 2: A comparative table of the previous approaches with the proposed one

Study	**Year**	**Dataset**	**Approach**	**Yaw Angle Range**	**Accuracy**
Shi, Liang, et al. [7]	2019	Multi-PIE, CASIA-WebFace	3DMM-aided pose-invariant model, deep neural network for feature extraction, and sparse representation-based classification	$-90°$ to $90°$	98 %
Zhang, Yi et al. [8]	2021	LFW	Pose-guided attention mechanism, deep neural network for feature extraction and classification	$-90°$ to $90°$	97.21%
Kang, Bong-Nam et al. [9]	2019	LFW	Pose-aware feature aggregation method, deep neural network for feature extraction	$-90°$ to $90°$	97.6%
Na, In Seop et al. [10]	2020	LFW	Deep residual attention network, deep neural network for feature extraction	$-90°$ to $90°$	97.51%
Lin, Wei-Chen et al. [11]	2023	Multi-PIE, CASIA-WebFace	Deep neural network for feature extraction, spatial attention mechanism	$-90°$ to $90°$	93.4%
Guan, Yepeng et al. [12]	2021	Multi-PIE, CASIA-WebFace	Deep neural network for feature extraction, clustering algorithm for pose normalization	$-90°$ to $90°$	95.6%
Ranjan, et al. [13]	2017	Multi-PIE, CASIA-WebFace	Deep neural network for simultaneous pose estimation and feature extraction	$-90°$ to $90°$	94.5%
Zhanfu, et al. [14]	2019	LFW, MegaFace	Deep neural network for feature extraction, adaptive normalization method	$-90°$ to $90°$	96.71%
Our proposed model	**2023**	**Taiwan Dataset**	**MTCNN, FaceNet, and SVC**	$-90°$ **to** $90°$	**99.174%**

5. Conclusion and Future Work

This study examined the impact of yaw pose variation on facial recognition accuracy and proposed a robust system optimized to detect faces with pose variations from $0° - \pm 90°$. The system employed MTCNN, FaceNet, and SVC and was trained and tested using the Taiwan dataset, which includes face images with varying yaw poses. The results demonstrate the effectiveness of the proposed system in detecting faces with pose variations and provide a valuable contribution to the field of facial recognition.

Future Work

1. Extend the proposed facial recognition system to detect faces with other types of pose variations, such as pitch and roll.
2. Investigate the impact of these additional pose variations on the accuracy of the system.
3. Optimize the system to detect faces with a wider range of poses and evaluate its performance on diverse datasets.
4. Analyse the trade-offs between accuracy, computational complexity, and memory usage.
5. Explore potential applications of the system in various fields, such as security, surveillance, and human-computer interaction.

References

[1] Ahmed N, Barczak ALC, Rashid MA, Susnjak T. A Parallelization Model for Performance Characterization of Spark Big Data Jobs on Hadoop Clusters. J Big Data. 2021;8:107.

[2] Sang G, Li J, Zhao Q. Pose-Invariant Face Recognition via RGB-D Images. Comp Intell Neurosci. 2016;2016:3563758.

[3] Lapuschkin S, Binder A, Muller KR, Samek W. Understanding and Comparing Deep Neural Networks for Age and Gender Classification. In: Proceedings of the IEEE international conference on computer vision workshops. 2017.

[4] Schroff F, Kalenichenko D, Philbin J. Facenet: A Unified Embedding for Face Recognition and Clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015:815-823.

[5] Kadirgama K, Rahman MM, Ghazali Z. Optimization of Heat Affected Zone for Air Plasma Cutting Process: A Support Vector Machine Approach. Int J Artif Intell Mechatron. 2013;2:13-18.

[6] Liu Y, Chen J. Unsupervised Face Frontalization for Pose-Invariant Face Recognition. Image Vis Comput. 2021;106:104093.

[7] Shi L, Song X, Zhang T, Zhu Y. Histogram-Based CRC for 3D-Aided Pose-Invariant Face Recognition. Sensors. 2019;19:759.

[8] Zhang Y, Fu K, Han C, Cheng P, Yang S, et al. PGM-Face: Pose-Guided Margin Loss for Cross-Pose Face Recognition. Neurocomputing. 2021;460:154-165.

[9] Kang BN, Kim Y, Jun B, Kim D l. Attentional Feature-Pair Relation Networks for Accurate Face Recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019: 5472-5481.

[10] Na IS, Tran C, Nguyen D, Dinh S. Facial UV Map Completion for Pose-Invariant Face Recognition: A Novel Adversarial Approach Based on Coupled Attention Residual UNets. Hum Centric Comput Inf Sci. 2020;10:1-17.

[11] Lin W, Chiu CTC, Shih KC. RGB-D Based Pose-Invariant Face Recognition via Attention Decomposition Module. ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE Publications. 2023; Vol. 2023.

[12] Guan Y, Fang J, Wu X. Multi-Pose Face Recognition Using Cascade Alignment Network and Incremental Clustering. Signal Image Video Process. 2021;15:63-71.

[13] Ranjan R, Patel VM, Chellappa R. Hyperface: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition. IEEE Trans Pattern Anal Mach Intell. 2019;41:121-135.

[14] An Z, Deng W, Hu J, Zhong Y, Zhao Y. APA: Adaptive Pose Alignment for Pose-Invariant Face Recognition. IEEE Access. 2019;7:14653-14670.

[15] Yao W, Varkarakis V, Lemley J, Corcoran P. On the Feasibility of Privacy-Secured Facial Authentication for Low-Power IoT Devices-Quantifying the Effects of Head Pose Variation on End-To-End Neural Face Recognition. In: Proceedings of the 24th Irish machine vision and image processing conference; August 2022:80-87.

[16] Wu Y, Zhang J, Fu H, Jin X. 'LPFF: A Portrait Dataset for Face Generators Across Large Poses.' 2023. arXiv preprint arXiv: https://arxiv.org/pdf/2303.14407.pdf

[17] Zhang L, Wang H, Chen Z. A Multi-Task Cascaded Algorithm With Optimized Convolution Neural Network for Face Detection Asia and the Pacific Conference on Communications Technology and Computer Science (ACCTCS). IEEE Publications. 2021.

[18] Li S, Lee HJ. Effective Attention-Based Feature Decomposition for Cross-Age Face Recognition. Appl Sci. 2022;12:4816.

[19] Yuan M, Nikouei SY, Fitwi A, Chen Y, Dong Y. Minor Privacy Protection Through Real-Time Video Processing at the Edge. In: 29th International Conference on Computer Communications and Networks (ICCCN). IEEE Publications. 2020;2020:1-6.

[20] Li HC, Deng ZY, Chiang HH. Lightweight and Resource-Constrained Learning Network for Face Recognition With Performance Optimization. Sensors. 2020;20:6114.

[21] Ding H, Zhou SK, Chellappa R. Facenet2expnet: Regularizing a Deep Face Recognition Net for Expression Recognition. 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE Publications; 2017.

[22] Darapaneni N, Evoori AK, Vemuri VB, Arichandrapandian T, Karthikeyan G, et al. Automatic Face Detection and Recognition for Attendance Maintenance. 15th International Conference on Industrial and Information Systems (ICIIS). IEEE Publications. IEEE Publications.2020.

[23] https://www.kaggle.com/datasets/omarabdulhalem/taiwan-dataset