FACE VERIFICATION USING CONVOLUTIONAL NEURAL NETWORK WITH PARTIAL TRIPLET LOSS ON FACE WEARING GLASSES

MUHAMMAD IKHSAN[1], RADEN SUMIHARTO[2], WAHYONO [3*]
[1] Computer Science Master Program, Universitas Gadjah Mada, Indonesia
[2,3] Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia
E-mail: [1]muhammadikhsan94@mail.ugm.ac.id, [2]r_sumiharto@ugm.ac.id, [3]wahyo@ugm.ac.id
(*corresponding author: Wahyono)

ABSTRACT

The role of face verification in the field of security and law enforcement has become an important part of daily life. However, the challenges involved in face verification still have an impact of inaccuracy on several factors that affect verification performance. One of them is the use of camouflage causing facial occlusion, such as wearing glasses. This research was conducted to find out the effect of wearing glasses on method performance. Therefore, CNN algorithm with siamese architecture is used to extract features from the face and propose to use the partial triplet loss function to partially minimize the optimization problem on the network. In experiment, our proposed approach achieves the lowest loss of 0.953 in training data, the highest accuracy of 68% for verification on face wearing glasses with partitioning and 73% accuracy for verification on face without wearing glasses with partitioning, both in verification data. These results are better than combination of CNN and transfer learning which only achieves 55%. Thus, it can be concluded that our proposed method could handle partial occlusion due to face wearing glasses.

Keywords: *Face Verification, Occlusion, Partial Image, Siamese, Triplet Loss*

1. INTRODUCTION

Face verification is the process of determining whether a pair of face images belong to the same or different subjects, which consists of facial feature extraction and face verification. The system will capture face objects and extract discriminatory features to represent the face image [1].

Face verification becomes a very useful and effective tool if key factors are met. Some key factors that can greatly affect the performance of facial verification are lighting levels, variations in the angle of rotation of the face, variations in facial expressions, and wearing glasses [2]. For face verification to function properly, it is very important that the overall face shape clearly shows all the features on the face. Full face verification is a technique of calculating the resemblance of a pair of facial images. In a simple comparison determining the facial resemblance, there are still some problems to verify the same person, one of which is the use of glasses that cause facial images to have partial occlusion [3]. This problem is difficult to handle because some features of the face are covered by glasses.

Research on the recognition or verification of the face wearing glasses has been developed [4]–[6]. Most of them perform facial recognition using Principal Component Analysis (PCA). The face image used is pre-processed by reducing the glasses attributes on the face image. Research [7, 8] use the Active Appearance Model (AAM) method to eliminate the glasses attribute on the face image. However, removing glasses from the face has an impact on the image becoming no longer authentic. To deal with occlusion problems on the face, it can be handled by dividing the image partially.

In this study, a partial image pre-processing approach was carried out. Several studies on partial image pre-processing were performed by dividing the image partially based on position of the eyes, nose, and mouth [9, 10]. Naveena et al. pre-process partial image using aspect ratios of images [11], while He et al. divided the image up and down; and left and right [12]. In this study, the image preprocessing approach is partially used and carried out differently. Our proposed method divided the image into several parts based on observation of each part size.

The accuracy of facial verification is influenced by the feature extraction algorithm used [13]. Song et al. used the PCA method which produces 83% accuracy by reconstructing facial images [4]. However, the PCA methods is computational expensive when producing the generation of eigenvalues and eigenvalues for matrix covariance. So that it is not useful for real-time implementation. Another method of face verification is performing convolutional neural network (CNN). Bukovcikova [14] and Koch [15] utilized CNN algorithm with siamese architecture which produced optimal accuracy. Wu et al. [16] produced high accuracy even close to perfect accuracy with utilization of CNN.

Generally, the last layer of the CNN could be used to find the commonality value between two objects with a distance function. Research in [17, 18] proposed triplet loss as a loss function by involving three facial images (anchor, positive, and negative), proven effective for face verification by applying the distance between pairs of faces with the same identity and different identities. However, when verifying the face wearing glasses, the method obtained lower accuracy due to occlusion.

Therefore, the contribution of the proposed research is for verifying the face image wearing glasses. Since the previous research did not consider the authenticity of facial images, in our method pre-processing is done by dividing the image partially with the same size. For each partial image, CNN algorithm with siamese architecture and triplet loss is then applied in order to extract feature of face images. Decision of verification will be carried out by calculating the average results of all partial images.

Finally, this paper will be divided into a number of sections. Section 2 discusses the detail of proposed method including design, algorithms, training, and testing. Section 3 provided detail implementation and experimental results, while Section 4 concludes our research.

2. THE PROPOSED METHOD

Our proposed method is divided into a number of steps including data acquisition, pre-processing which includes face data, dividing the image into several sub-regions (partial images), converting RGB images to grayscale, and changing the image size of all partial data of the same size. Furthermore, for extracting the feature of each sub-region image, CNN algorithm is performed. In order to calculate the error rate, triplet loss is used to overcome optimization problems in the network.

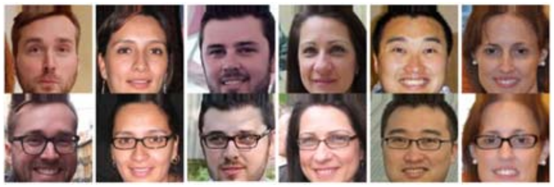

Figure 1: Selected sample from dataset MeGlass [19]

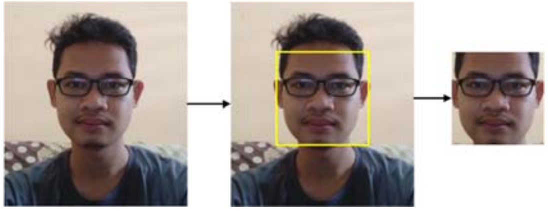

Figure 2: Illustration of face detection

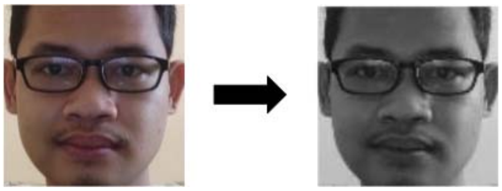

Figure 3: Illustration of image conversion

Finally, several sets of tests were conducted including testing of sub-regions, number of regions, comparison of images using glasses and without glasses, and comparison of other architectural models.

2.1 Data Acquisition

The data used in the experiment is consist of both face wearing glasses and face without glasses images. These images are obtained from MeGlass Dataset [19] consisting of 1,538 identities with a total of 46,817 face images. Each face image has 120×120 size and RGB color. However, due to resource limitations, 1,000 identities were used with total of 32,000 face images. This data is then randomly divided into training and testing data. The training data is composed by 20,000 pairs of samples, while the testing data contains 12,000 pairs of samples. Each identity consists of the image of a face wearing glasses and without glasses. A sample MeGlass dataset is shown in Figure 1.

2.2 Preprocessing

Before applying face verification stage, the input image will go through the data pre-processing stage first, so that data processing is easier to be performed. The preprocessing is divided into several stages, including face detection, RGB-grayscale conversion, partially face images partition, and finally resizing the image to a size adjusted to the model input of CNN.

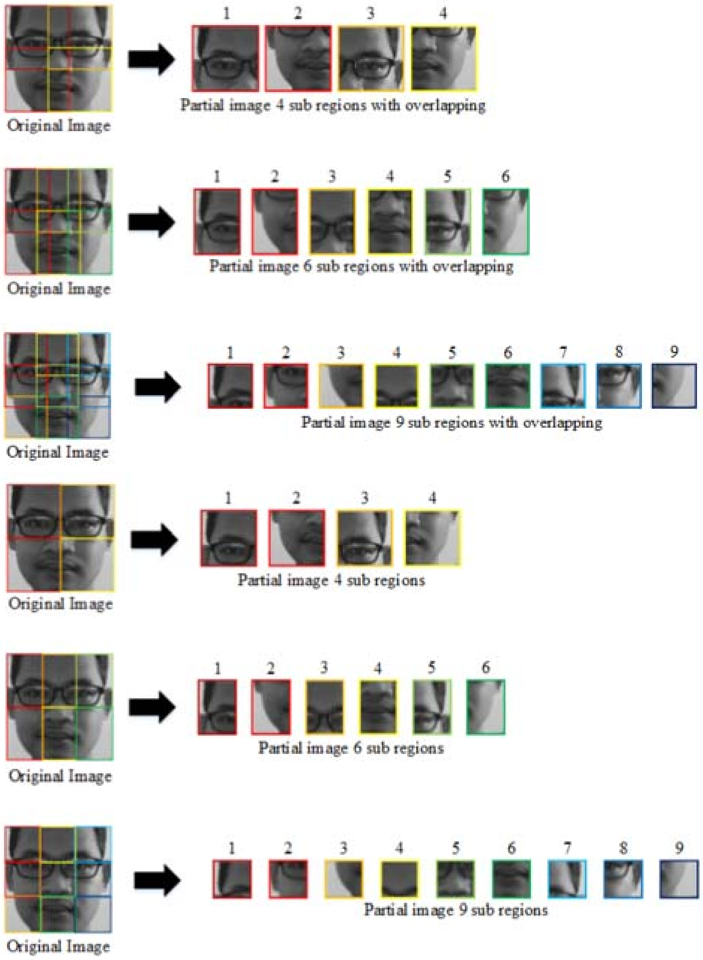
Original Image
1
2
3
4
Partial image 4 sub regions with overlapping
Original Image
1
2
3
4
5
6
Partial image 6 sub regions with overlapping
Original Image
1
2
3
4
5
6
7
8
9
Partial image 9 sub regions with overlapping
Original Image
1
2
3
4
Partial image 4 sub regions
Original Image
1
2
3
4
5
6
Partial image 6 sub regions
Original Image
1
2
3
4
5
6
7
8
9
Partial image 9 sub regions
Original Image

Figure 4: Illustration of partial image without overlapping and with overlapping.

2.2.1. Face Detection

The initial pre-processing stage is face detection. In this stage, a Haar feature-based cascade classifier method is used to separate face regions from non-face regions [20]. This method works by extracting features from faces and it has the advantage of rejecting images that do not contain face region, so that in the next step, we only focus on face region in order to apply face verification. Then, the image is cropped according to the size of the detected face region. Illustration of face detection using the Haar feature-based cascade classifiers method is shown in Figure 2.

2.2.2. Image Conversion

The next pre-processing stage is image conversion. Conversion of image colors from RGB (Red, Green, Blue) to grayscale is needed to reduce the number of channels to be processed. The process of image conversion from RGB to grayscale is shown in Eq. (1). An example illustration of image conversion from RGB to grayscale is shown in Figure 3.

$$grayscale = w_R R + w_G G + w_B B \qquad (1)$$

2.2.3. Image Partitioning

After the face image is converted to grayscale color, then the image will be divided into 4, 6, and 9 sub-regions with overlapping and without overlapping approaches. These approaches are done for finding the number of certain sub-regions that provides optimal accuracy. Illustration of partial images partition with and without overlapping are shown in Figure 4.

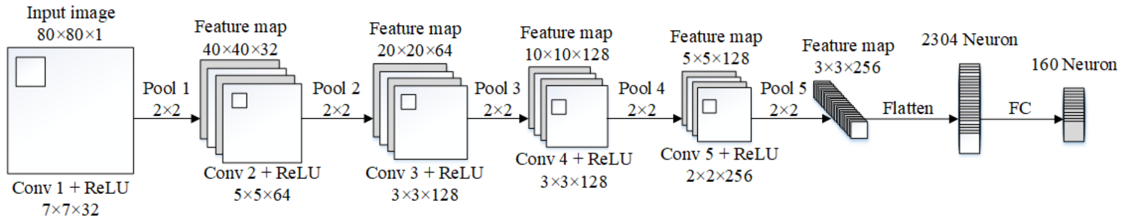

Figure 5: The proposed CNN architecture.

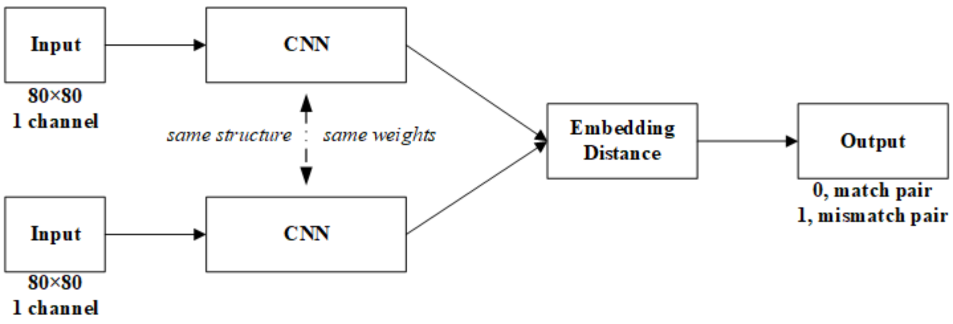

Figure 6: The proposed CNN-Siamese architecture.

The images of partial division for 4, 6, and 9 sub-regions with and without overlapping produce different image size for each region. Each region of image partition without overlapping has a size of 60×60 pixels for 4 sub-regions, 40×60 pixels for 6 sub-regions, and 40×40 pixels for 9 sub-regions. In the other hand, each region of images partition with overlapping has a size of 72×72 pixels for 4 sub regions, 48×72 pixels for 6 sub regions and 48×48 pixels for 9 sub regions.

2.2.4. Image Resizing

In this process, because face images that have been partially divided into 4, 6, and 9 sub-regions in both with and without overlapping have different size, each partial image is then resized according to the input size in the CNN model. In our research, we proposed to resize all partial images becomes 80×80 pixels, so that each partial image has the same size to be used as input in CNN model.

2.3 CNN with Siamese Architecture

In this research, CNN with siamese architecture is used as feature extraction, based on architecture proposed by Bukovcikova [14]. This CNN network consists of 5 convolution layers and 5 subsampling layers. It has an 80×80 pixels in input layer and uses a 7×7 kernels on the first convolution layer with stride 1. The next convolution layer gradually getting smaller such as 5×5, 3×3, 3×3, and 2×2, respectively. In addition, Rectified Linear Unit (ReLU) is used as an activation function for each convolution network, as it provides faster computing time comparing other activation functions. The design of the CNN algorithm is shown in Figure 5, while the design of Siamese network is shown in Figure 6.

The first 2 convolution layers are always followed by 2 subsampling layers, namely max pooling with 2×2 kernels, 2 strides, zero mean with a standard deviation of 0.01 [21]. The input layer is a grayscale image with pixel values between 0 and 255 which later should be normalized into values between 0 and 1. This normalization process is carried out to reduce the computing process. The output from the last convolution layer is transformed into a vector and used as input into the fully connected layer. Tanh is used as an activation function (2) and output normalization (3) is performed to normalize the output values in the range of 0 to 1.

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (2)$$

$$S(x) = \sum_{i} \frac{\max(x)-\min(x)}{x_i-\min(x)} \quad (3)$$

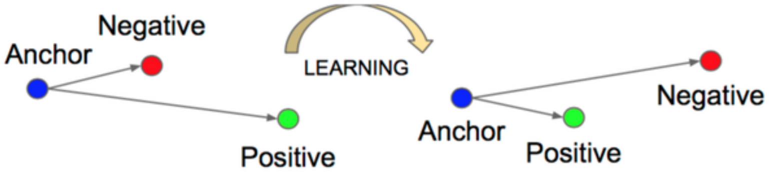

Figure 7: Illustration of triplet loss application.

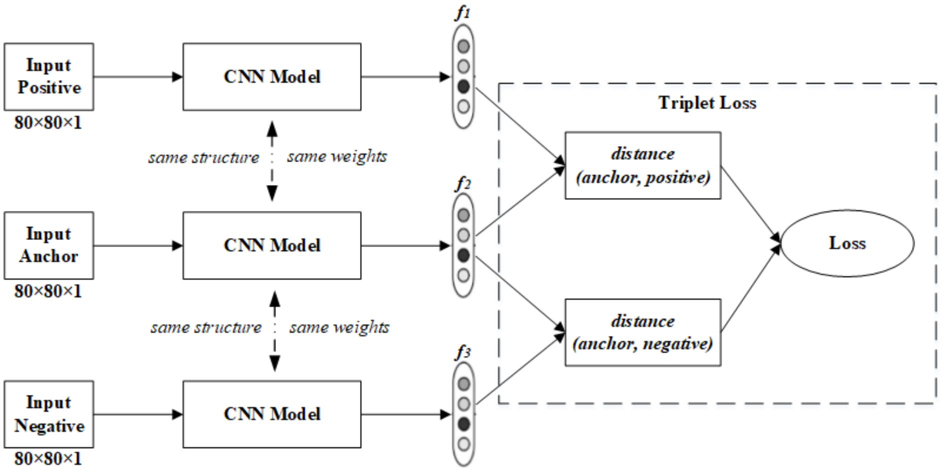

Figure 8: The design of triplet loss application.

The design of the CNN algorithm in Figure 5 is initialized with an input image size of 80×80×1 which changes the size of the previous study which is 52×72×3 [14]. However, it does not change other hyperparameters in the convolution layer. Changing input size of network affects the number of parameters trained. The image input size of 80×80×1 produces 774,432 training parameters, while the size of 52×72×3 produces 655,010 training parameters.

The training data contains 20,000 pairs of images randomly by paying attention to 2 positive anchor images that are the same identity and negative is a different identity. On the other hand, test data contains 12,000 pairs of images randomly in a ratio of 1:1 between positive and negative classes. The hyperparameters used in this research are shown in Table 1.

Table 1: Variations in Hyperparameter Tuning Experiments

Hyperparameter	Variations
Learning Rate	0.0001
Dropout	0.5
Weights	Random (0, σ = 0.01)
Bias	0.5
Batch Size	32, 64, 128
Epoch	20, 50, 70

2.4 Similarity Measurement using Triplet Loss

Output vectors are usually called embedding, where the embedding of an image is inserted into the euclidean space. In euclidean space, the pair of images used as input will produce a similarity distance. In this study, triplet loss is used to measure distances between face objects and minimize optimization problems that result from a pair of faces. The use of triplet loss is motivated from [17, 18] by applying it to face recognition. By involving 3 input images, such as anchor and positive images as faces with the same identity, while negative images as faces with different identities and margins as hyperparameters are added to widen the distance from the same and not the same pair. Applying margins between pairs of faces with the same identity and faces with different identities, triplet loss keeps the distance of faces with the same identity closer than faces of different identities in the embedding space as shown in Figure 7. The design of triplet loss application in this study is shown in Figure 8 and the triplet loss formula is shown in Eq. (4).

$$Loss = max\left(\|f_a - f_p\| - \|f_a - f_n\| + margin, 0\right) \quad (4)$$

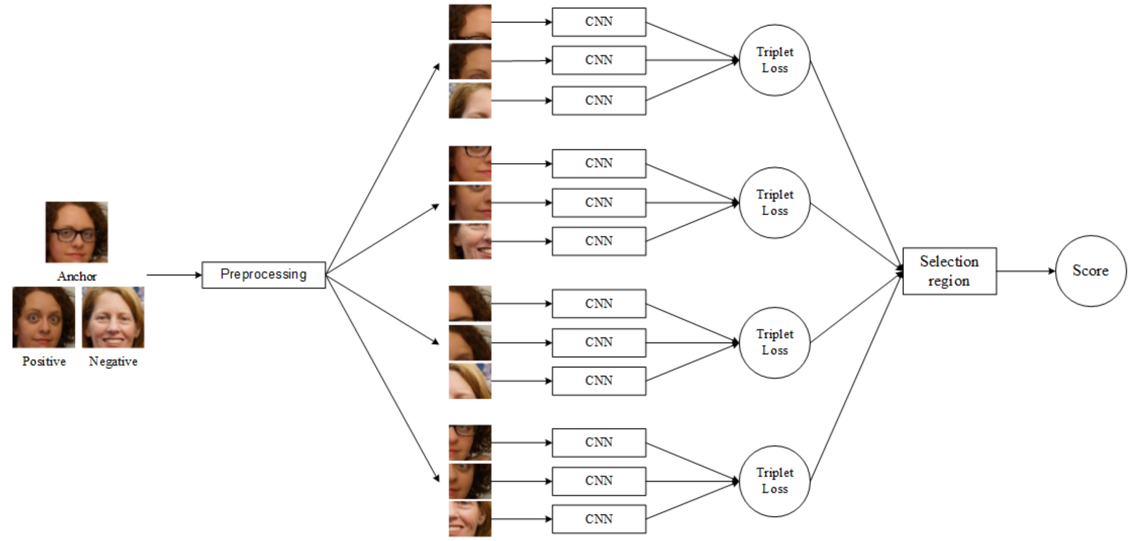

Figure 9: Illustration of triplet loss application.

where ||fa – fp|| is the distance between the anchor and the positive images which are the same identity, ||fa – fn|| is the distance between the anchor and the negative images which are different identity, and margin is hyperparameter used to widen the distance between positive and negative classes.

Measuring the accuracy of face verification based on triplet loss is shown in Eq. (5):

$$Accuracy = \|f_a - f_p\|^2 + margin \leq \|f_a - f_n\|^2 \quad (5)$$

Training data consisting of several sub-regions is utilized. Each sub-region then produces a loss that is used for handling optimization problems in the network. Output in the form of vector space from the extracted image is then used to compare the distance between the paired images. The resulting distance from the anchor-positive image is reduced by the distance of the anchor-negative image and added to the specified margin. Training on subregion images using triplet loss is carried out on each subregion image. Parts of the sub-region image are extracted and the loss is calculated. Then, a loss selection is made for regions where there is no partial occlusion, if there are two or more regions where there is no partial occlusion, the average value of the loss is calculated. An illustration of the partial triplet loss process is shown in Figure 9.

2.5 Training Strategy

The training design consisted of several steps, including hyperparameter tuning experiments and model training.

2.5.1. Hyperparameter Tuning

Hyperparameter tuning experiments were conducted to find hyperparameters that provide optimal results. The CNN model used in Figure 5 and the hyperparameter variations used are in Table 1. In Table 1, there are several parameters carried out by the experiment to be determined as an optimal hyperparameter. These hyperparameter are learning rate, dropout, weights, and bias values. In the experiment, it is found that the optimal hyperparameter learning rate is 0.0001 because it results in an increase in deep learning compared to the value of a large learning rate. The choice of a dropout value of 0.5 which is only used in the fully connected layer was chosen because it avoids overfitting by removing 50% of the neurons. Weights and biases used refer to [21] by applying weights with a zero average Gaussian distribution with a standard deviation of 0.01 and the bias is fixed at each layer that is 0.5. Furthermore, other hyperparameters to be determined by experiment were batch size and epoch. The combination of these two hyperparameter will be analyzed in order to determine optimal accuracy. Batch size determines the number of samples in each mini batch. Batch size adjustments are made to adjust between two differences, namely accurate gradient direction, or iteration speed. An epoch tuning experiment was carried out to determine the effect of the number of epochs to achieve local optima.

2.5.2. Model Training

The next stage is to train the model using the siamese network architecture with hyperparameters obtained from previous stages e.g. hyperparameter tuning experiments. The inputs used in the training model are adjusted to computational resources, while the output of the network model will be accuracy and average time of training per epoch. To calculate the distance from the pair image, we utilized the Euclidean distance, while the accuracy of model will be computed by using Eq. (5).

Beside the accuracy, the model also produces a loss function in the learning process. The loss values generated in a model is obtained using the triplet loss method. The use of triplet loss method is done with the aim to minimize optimization problems in learning. The loss value generated from the triplet loss method uses Eq. (4), where the distance of the image of a couple with the same identity is closer than the image of a couple with a different identity. The lower of loss produced, the better the learning model is obtained. The triplet loss method has a single hyperparameter, i.e. margin. Initialization margins in the triplet loss method are generally set for each measurement of similarity to be 0.2.

2.6 Testing Design

Verification testing is done to prove whether the proposed model built using siamese network and triplet loss gives optimal accuracy. Architectural models used for data testing is presented in Figure 6. The testing data is randomly selected from a collection of paired images then their features are extracted using the CNN algorithm. The resulting output is calculated using Euclidean distance to measure the similarity of face object. The result of the similarity distance is normalized so that the low distance has a similarity level to the positive while the high distance to the negative.

2.6.1. The Effect of Number of Sub-regions

The test is carried out by comparing face images that have been divided into 4, 6, and 9 sub-regions with and without overlapping approaches. The purpose of this test is to determine the sub-region with a certain size that provides optimal accuracy compared to the others.

2.6.2. The Effect of Glasses

Test is carried out on the face image wearing glasses and without glasses using the siamese network algorithm and triplet loss. The purpose of his test is to prove that the siamese network algorithm and triplet loss provides optimal accuracy in both face image wearing glasses and without glasses.

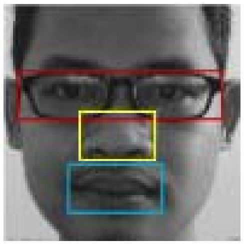

Figure 10: Illustration of facial images and their features.

2.6.3. Comparison with Other Methods

This test compares the architecture of the proposed model with the architecture of other models. The purpose of this test is to compare model architectures that provide optimal results in face verification tasks.

2.7 Evaluation

The face has several important features, including eyes, nose, and mouth. Partial occlusion caused by the use of glasses causes masking or obscuring one of the important features of the face, the eyes. However, other features such as the nose and mouth do not have partial occlusion. The three main features are used as a benchmark to determine the similarity of two similar or different facial images. Illustrations of facial images and their features are shown in Figure 10.

In the verification test stage for face wearing glasses, the region that produces features without partial occlusion becomes a reference to determine the accuracy of the training or the accuracy of verification. As in Figure 4, partial images of 4 sub-regions without overlapping and with overlapping that produce features without partial occlusion, namely region 2 and region 4, partial image with 6 sub-region, region 4, partial image with 9 sub-region, region 5 and region 6. If there are two or more regions to determine accuracy, then take the average accuracy of the sum of these regions. Whereas for verification testing without wearing glasses, the output of each region is summed, and the average results are taken.

Table 2: Results of Hyperparameter Tuning Accuracy And Loss Experiments

Batch Size	Epoch 20		Epoch 50		Epoch 70	
	Acc	Loss	Acc	Loss	Acc	Loss
32	0.455	5.128	0.688	6.865	0.682	6.166
64	0.402	6.547	0.644	9.724	0.708	10.006
128	0.346	21.202	0.638	20.188	0.667	36.543

Table 3: Result of Accuracy and Loss for Model Training

Data	Accuracy	Loss
Without Partial	0.860	3.667
4 Sub Region	0.808	4.092
6 Sub Region	0.846	0.953
9 Sub Region	0.791	4.307
4 Sub Region + Overlapping	0.835	5.774
6 Sub Region + Overlapping	0.855	1.535
9 Sub Region + Overlapping	0.834	5.190

Table 4: Result of Each Region Accuracy of Partial Image

	Without Overlapping			With Overlapping		
	4	6	9	4	6	9
Region 1	0.790	0.748	0.636	0.829	0.792	0.709
Region 2	0.807	0.714	0.753	0.852	0.796	0.808
Region 3	0.775	0.824	0.644	0.825	0.833	0.743
Region 4	0.808	0.846	0.730	0.819	0.855	0.753
Region 5	-	0.756	0.821	-	0.776	0.849
Region 6	-	0.743	0.762	-	0.448	0.819
Region 7	-	-	0.692	-	-	0.730
Region 8	-	-	0.761	-	-	0.784
Region 9	-	-	0.685	-	-	0.744

3. EXPERIMENTAL RESULTS

3.1 Hyperparameter Tuning Experiment

In applying the CNN model with the siamese architecture, there are several hyperparameters that affect the accuracy in the model including batch size and epoch as shown in Table 1. The results of the batch size and epoch experiments are shown in Table 2. As depicted in Table 2, batch and epoch sizes give effect to the accuracy and loss values. The batch size of 32 gains the highest accuracy in the epoch 50 which obtain accuracy of 68%, when the batch size value was increased to 64 the accuracy was 70% in the epoch 70 and when the batch size was increased again to 128 the accuracy decreased to 66% in the epoch 70 experiment. On the other hand, it is found that the batch size 32 produces lower loss value compared to batch sizes 64 and 128, even the accuracy produced is not too significant. Therefore, batch size 32 and epoch 50 were chosen as optimal hyperparameters in both training and testing stages.

3.2 Model Training

Model was trained using both face images with and without partition. The hyperparameter used in this stage is shown in Table 1 and the results of the tuning experiment are shown in Table 2. The training data consists of 20,000 pairs of face images, of which 16,000 pairs of images are used for training and 4,000 pairs of images for validation. The accuracy results are shown in Table 3. As shown Table 3, it is found that the accuracy of training data on whole images produces training accuracy of 86% while the partial data can produce the highest accuracy on data with a smaller number of partials. In partial data training, the number of partials with 4 sub regions produces the highest accuracy compared to 6 and 9 sub regions in both with and without overlapping approaches. From these results, it is concluded that 4 sub-regions produce better accuracy compared to the others.

Beside accuracy, the training stage also produces a loss on the model. The loss generated by the model is calculated by the average loss obtained from the model training using the triplet loss method. As results of Table 3, it is noted that loss function training on partial data can produce the lowest loss function on data with a partial number of 6 sub regions. The loss produced in partial data training is better than training in whole image data, although the accuracy produced is not the best but the error rate in learning is better.

If examined deeper to measure the accuracy of the model, the accuracy of each region in the partial image training is shown in Table 4. Table 4 shows that certain regions produce better accuracy than other regions. This is because certain regions do not have unique features that distinguish the same face or there are partial occlusions in the face image. By observing results, in the training of 4 sub regions, testing region 2, and region 4 gain better accuracy compared to region 1 and region 3. In region 1 and region 3 there is a partial occlusion which causes accuracy to be decreased than in region 2 and region 4 which are not have partial occlusion.

In the training data for 6 sub regions, region 4 produces better accuracy compared to other regions. The decision to determine region 4 as the accuracy of the model in the 6 subregions training is because the region has two features in the image namely nose and mouth, as shown in Figure 4. The same thing happened in the 9 subregions training where region 5 and region 6 produce better accuracy compared to other regions. The effect of partial images using overlapping improves training accuracy in all partial image regions.

3.3 Evaluation on Data Testing

This testing uses a model that has been trained with a predetermined hyperparameter in the previous stage. The step of calculating the accuracy in the sub-region data is determined by calculating the similarity distance using the Euclidean distance in each sub-region.

The resulting distance is added according to the partial image from the same image, then the average is taken. The threshold used to determine positive and negative classes is iterated in each test. The verification of the dataset without glasses was also added to determine the effect of using glasses on the accuracy of verification. Dataset without glasses in the study consists of 12,000 pairs of images [22].

As shown in Table 5, it is found that face wearing glasses affects the accuracy of verification. Figure 11 provides a general description of all the true cases and Figure 12 provides a general description of all the failed cases. Testing stage experiments were conducted on the variation of the utilization of glasses and without glasses and it produced the highest accuracy of 73%. In addition, the number of image partition also gives effect on the accuracy of face verification. This is because some partial images have unfavorable facial features. Therefore, in testing some partial number, a small part of the partial image of the face has a level of accuracy that is not optimal. The decrease in accuracy also occurs in the verification of faces wearing glasses. This is because the test does not involve regions that have partial occlusion. So that if the accuracy is calculated based on the average value of all sub-regions, it will produce less accuracy. This affects to the distance of similarity if it is matched to the face image without glasses.

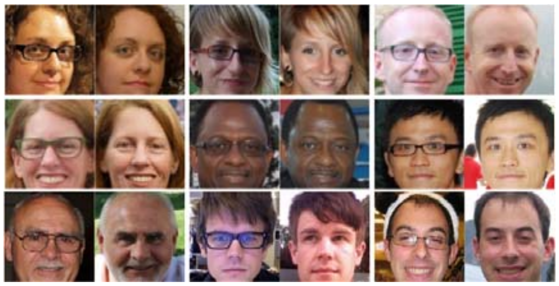

Figure 11: True positive samples

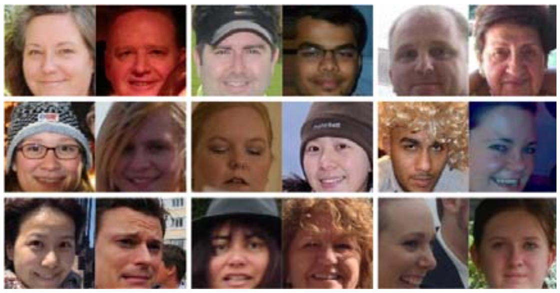

Figure 12: False positive samples

In the face verification test without glasses, the highest accuracy was 89%. This test also has a partial effect on accuracy. The level of accuracy is decreased or increased which is not too significant in testing certain regions. Increased accuracy occurs in testing the partial image of 4 overlapping sub-regions.

To find out how well the proposed model comparing to other models, we compared the AUC rates of the models. The model with the highest AUC rate produces better precision because it has a higher rate of true positives, or conversely has lower false positives. The comparison of the AUC for wearing glasses and without glasses is shown in Figure 13.

Table 5: Result of Comparison Accuracy Verification Wearing Glasses and Without Glasses

Dataset	Without Partial	Partial Without Overlapping			Partial with Overlapping		
		4	6	9	4	6	9
Variation of Glasses and Without Glasses	0.733	0.652	0.619	0.593	0.684	0.573	0.589
Without Glasses	0.880	0.868	0.886	0.848	0.894	0.844	0.879

Table 6: Comparison with Other Architectural Models

Architecture	Image Size	Without Partial	Partial Without Overlapping			Partial with Overlapping		
			4	6	9	4	6	9
Proposed Method	80×80×1	0.733	0.652	0.619	0.593	0.684	0.573	0.589
Siamese Net [16]	256×256×3	0.733	0.615	0.640	0.653	0.650	0.674	0.659
CNN + TL [18]	160×160×3	0.553	0.544	0.545	0.552	0.539	0.550	0.555

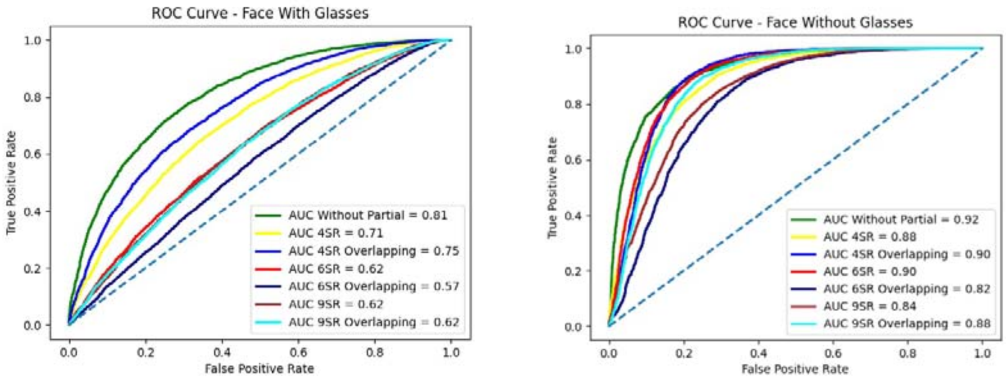

Figure 13: The AUC results of verification testing on face wearing glasses and without glasses.

Testing on a model that wearing glasses without the partitioning process produces an AUC level of 0.81, while testing on a model with a partition process produces the highest AUC level of 0.75. In contrast, testing on the model without glasses and without the partitioning process yielded an AUC level of 0.92, while testing on a model with a partitioning process resulted in the highest AUC level of 0.90. From these results, it can be concluded that the model wearing glasses without partitioning and the model with partitioning has a significant difference in accuracy. In contrast, testing on model without glasses produced almost the same AUC levels, in both with and without the partitioning process.

3.4 Comparation with Other Architecture

Comparison with other model architectures is carried out to determine whether the proposed model architecture is better than the existing model architecture [23]. The model architecture that becomes the reference is the siamese network model architecture proposed by Wu et al. [16] and the CNN triplet loss model architecture proposed by Schroff et al. [18]. Each model architecture in this comparison is first trained using the same hyperparameter as the proposed model architecture.

As shown in Table VI, it is found that the highest accuracy is 73% and the results obtained for the proposed model and the Wu model architecture have the same accuracy. However, the proposed model is better because the training parameters are less so that it increases the speed of the model in training and predicting data. Meanwhile, testing the Schroff model architecture produces an accuracy of 55%.

Apart from testing the model without partitioning, the model architecture is also used to compare the model with the partitioning process. In this test, an experiment was carried out to determine the effect of partial images on each model on the accuracy value. Partial image testing on each model architecture is evaluated by selecting a region that produces an image with facial features without partial occlusion. As shown in Figure 9, the facial features selected are the nose and mouth. The selected regions in the partial image of 4 sub-regions are region 2 and region 4, then the partial image of 6 sub-regions is region 4, and the partial image of 9 sub-regions is region 5 and region 6.The results of the selected region are as shown in Table 6.

As shown in Table VI, the proposed model is superior in testing the partial image of 4 sub-regions without overlapping or overlapping. While testing on other partial models, the model architecture [16] provides better accuracy than other models.

The parameters for the proposed model are fewer than the parameters for the other models, and the image size used for each model is different. The resizing of the image then produces an image blur effect on the proposed model of 0-100%, model [16] of 110-540%, and model [18] of 30-300%. Therefore, the large image blur effect makes it difficult for the model to detect the features clearly and in detail. This will have an impact on the number of false positives or false negatives.

4. CONCLUSION

The CNN siamese and triplet loss methods were successfully applied for face verification wearing glasses with AUC values of 68% in partial data and 75% in whole image data. The results of lost values in partial image training are better than the image without partial. Verification test with glasses without glasses was added to find out the comparison and the effect of verification accuracy on the face image wearing glasses. Using our proposed approach, this problem has been solved as our method achieve accuracy of 73%, which is better than method of combination CNN and Transfer Learning of 55%.

Future work will focus more on the data used, in this study there are still some side view faces so that when the results of partial processing of facial images are not evenly distributed. Therefore, in further development it is expected to apply augmentation so that the side-view image can be improved to the front view. Furthermore, the effect of the blur effect on images are too large causes the image to appear unclear and not detailed when extracting features so that verification accuracy is low. It takes a feature extraction model that produces better accuracy with the input image size is not too large. Finally, focusing on the model by exercising more variation in the data so that the resulting model is better, and the loss obtained is lower.

ACKNOWLEDGMENT

The researchers are grateful to Universitas Gadjah Mada for providing financial assistance for this research partially through the Final Assignment Recognition Program (RTA) Year 2020 with Grant Number 2488/UN1.P.III/DIT-LIT/PT/2020.

REFRENCES:

[1] R. Yan, Z. Zhong, J. Zhang, and Y. Xu, “An improved similarity metric based on joint Bayesian for face verification,” in 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, Dec. 2016, pp. 222–226, doi: 10.1109/ICCWAMTIP.2016.8079842.

[2] A. Al-Azzawi, H. Al-Sadr, J. Cheng, and T. X. Han, “Localized Deep Norm-CNN Structure for Face Verification,” in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, Dec. 2018, pp. 8–15, doi: 10.1109/ICMLA.2018.00010.

[3] F. Qiu, S. Kamata, and L. Ma, “Deep Face Recognition under Eyeglass and Scale Variation Using Extended Siamese Network,” in 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), Nanjing, China, Nov. 2017, pp. 471–476, doi: 10.1109/ACPR.2017.48.

[4] C. Song, B. Yin, and Y. Sun, “Eyeglasses Eigenface Based Glasses-face Recognition,” in 2008 IEEE International Conference on Networking, Sensing and Control, Sanya, China, Apr. 2008, pp. 1385–1390, doi: 10.1109/ICNSC.2008.4525435.

[5] G. Pei and S. Fei, “Enhanced PCA reconstruction method for eyeglass frame auto-removal,” in 2014 4th IEEE International Conference on Network Infrastructure and Digital Content, Beijing, China, Sep. 2014, pp. 359–363, doi: 10.1109/ICNIDC.2014.7000325.

[6] T. Hosoi, S. Nagashima, K. Kobayashi, K. Ito, and T. Aoki, “Restoring occluded regions using FW-PCA for face recognition,” in 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,

Providence, RI, USA, Jun. 2012, pp. 23–30, doi: 10.1109/CVPRW.2012.6239211.

[7] Y.-K. Wang, J.-H. Jang, L.-W. Tsai, and K.-C. Fan, “Improvement of Face Recognition by Eyeglass Removal,” in 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Darmstadt, Germany, Oct. 2010, pp. 228–231, doi: 10.1109/IIHMSP.2010.64.

[8] J. Heo and M. Savvides, “Face Pose Correction With Eyeglasses and Occlusions Removal,” in 2007 Biometrics Symposium, Baltimore, MD, USA, Sep. 2007, pp. 1–6, doi: 10.1109/BCC.2007.4430551.

[9] S. Nikan and M. Ahmadi, “Partial Face Recognition Based on Template Matching,” in 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Bangkok, Thailand, Nov. 2015, pp. 160–163, doi: 10.1109/SITIS.2015.19.

[10] A. Elmahmudi and H. Ugail, “Experiments on Deep Face Recognition Using Partial Faces,” in 2018 International Conference on Cyberworlds (CW), Singapore, Oct. 2018, pp. 357–362, doi: 10.1109/CW.2018.00071.

[11] M. Naveena, G. HemanthaKumar, Prakasha M, and P. Nagabhushan, “Partial face recognition by template matching,” in 2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, Dec. 2015, pp. 319–323, doi: 10.1109/ERECT.2015.7499034.

[12] L. He, H. Li, Q. Zhang, and Z. Sun, “Dynamic Feature Matching for Partial Face Recognition,” IEEE Transactions on Image Processing, vol. 28, no. 2, pp. 791–802, Feb. 2019, doi: 10.1109/TIP.2018.2870946.

[13] B.-S. Oh, K.-A. Toh, K. Choi, A. Beng Jin Teoh, and J. Kim, “Extraction and fusion of partial face features for cancelable identity verification,” Pattern Recognition, vol. 45, no. 9, pp. 3288–3303, Sep. 2012, doi: 10.1016/j.patcog.2012.02.027.

[14] Z. Bukovcikova, D. Sopiak, M. Oravec, and J. Pavlovicova, “Face verification using convolutional neural networks with Siamese architecture,” in 2017 International Symposium ELMAR, Zadar, Sep. 2017, pp. 205–208, doi: 10.23919/ELMAR.2017.8124469.

[15] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese Neural Networks for One-shot Image Recognition,” p. 8, 2015.

[16] H. Wu, Z. Xu, J. Zhang, W. Yan, and X. Ma, “Face Recognition based on Convolution Siamese Networks,” in International Congress on Image and Signal Processing, Shanghai, China, Oct. 2017, p. 5

[17] Z. Ming, J. Chazalon, M. M. Luqman, M. Visani, and J.-C. Burie, “Simple Triplet Loss Based on Intra/Inter-Class Metric Learning for Face Verification,” in 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Oct. 2017, pp. 1656–1664, doi: 10.1109/ICCVW.2017.194.

[18] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, Jun. 2015, pp. 815–823, doi: 10.1109/CVPR.2015.7298682.

[19] J. Guo, X. Zhu, Z. Lei, and S. Z. Li, “Face Synthesis for Eyeglass-Robust Face Recognition,” arXiv preprint arXiv:1806.01196, p. 11, 2018.

[20] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” International Journal of Computer Vision, vol. 57, pp. 137–154, 2004, doi: https://doi.org/10.1023/B:VISI.0000013087.49260.fb.

[21] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the Gap to Human-Level Performance in Face Verification,” in 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, Jun. 2014, pp. 1701–1708, doi: 10.1109/CVPR.2014.220.

[22] M. Weber, “The Caltech Frontal Face Database,” Computational Vision, California Institue of Technology, USA, 1999. http://www.vision.caltech.edu/html-files/archive.html (accessed Mar. 25, 2020).

[23] M. Ikhsan, “Face Verification Using Partial Triplet Loss on Face Wearing Glasses,” Master of Computer Science Thesis, Dept. Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, 2020.