Facial Recognition Through Siamese CNNs

Leenane Tinashe Makurumure
University Of KwaZulu-Natal
217076701@stu.ukzn.ac.za

January 10, 2021

Abstract

Facial images offer a reliable means of physical biometric identification. However, image modularities, geometric distortions, partial deformations, and obstructions could easily lead to false positives or false negatives. This paper explores the viability of a Patch-based Siamese CNN for region-specific extraction in situations described above. The idea is to create a facial recognition model that will still perform even when only partial facial information is available. We extract nine patches and create nine Patch-specific models. We explore the accuracy of three Patch-based models that uses different combinations of Patch-specific sub-models against one that uses a global facial image. Training and testing are performed on the AT&T face dataset. Experimental work shows that carefully combined Patch-Specific CNNs can perform better than a Global CNN. The Global CNN classified image pairs with an EER of 0.090. A Patch-based Siamese CNN of all nine patches achieved an EER of 0.045. Two Patch-based Siamese CNNs, one with carefully chosen Patch-specific sub-models and the other with random Patch-specific sub-models, achieved an EER of 0.037 and 0.098, respectively.

I. Introduction

To date, facial recognition has been successfully incorporated into simple social media applications as a means of object identification. However, it is not reliable enough for biometric identification or authentication to access a system or some resource. Facial recognition could offer a relatively cheap solution to biometric authentication, as illustrated in Figure 1. However, it is not very reliable, and this has kept it at the forefront of research in the past decade. There have been vast improvements due to the advancement of deep learning and faster processing of enormous data. Nevertheless, there is still more to be done. Ken Bodnar, an AI researcher, is quoted in an article [17] saying that AI facial recognition technology is excellent but not very robust. This means that the technology could misidentify someone as a genuine client (False acceptance) or an imposter (False rejection), thereby causing a breach into the system or inconveniencing legitimate clients, respectively.

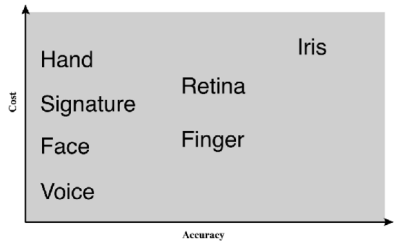

Figure 1: *Cost versus accuracy of various biometric characteristics in user authentication schemes. Adapted from [16].*

A study from MIT shows that facial recognition tools had significant problems identifying people of color [17]. Bias and potential invasion of privacy have kept facial recognition as an essential research topic. Facial images are typically required in two different electronic tasks: verification and identification, as shown in Figure 2. Verification (authentication) performs one-to-one matching, while identification is a

one-to-many matching problem. In both cases, the underlying objective matches a test image (known or unknown) to another image to determine if both images belong to the same person or are from different people.

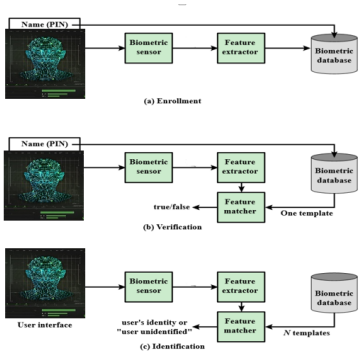

Figure 2: *A Generic Biometric System. Enrolment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user, or identifying an unknown user [16].*

Conventional strategies use classifiers like neural networks (NN) or support vector machines (SVM) to measure similarities between two images and classify them as the same person or imposter using some threshold. However, these methods have limitations in facial recognition due to the sheer volume of data that requires processing before a decision can be made. Better results of neural networks applied to the facial recognition problems have come from siamese convolutional neural networks. Convolutional networks (CNNs) are non-linear, multi-layer neural networks that operate pixel by pixel to learn low-level features and high-level representations in a unified way. Past research techniques tend not to perform any significant pre-processing to the image pairs, and features are typically extracted from the entire facial image (global CNNs). They do not address the shortcomings that come up when the facial image is disfigured or deformed. We propose an improvement to the conventional siamese CNN architecture by creating a neural network whose input is not the whole face but patches/regions of the face (Patch-based siamese CNN). This technique would allow the network to specialize on a smaller area and is less likely to give a false acceptance or false rejection when the entire face is unavailable. This research aims to design and implement techniques and experiments to evaluate the viability of Patch-based Siamese CNNs for facial image matching.

This study seeks to answer the following research questions: Can Patch-based feature extraction be more effective that global feature extraction when applied to images taken in real environments? Can less information (partial face) be used instead of the entire face and still get the same or better results.

II. Literature Review

Facial verification studies, prior to 2014, generally compared features extracted from two faces separately before the idea of a siamese architecture. The concept of siamese architecture was first introduced by [2], who applied it to signature verification. Siamese neural network is an architecture of twin neural networks with identical weights that take different inputs and work simultaneously to compute similar output vectors. Chopra [4] replaced the Siamese neural network subnets with CNNs and applied it to face verification. The CNNs map input patterns into a low-dimensional target. This idea started with the PCA-based eigenface method [19], which is invariant to geometric distortions and small differences in input pairs. This drives the computing of a similarity metric between the patterns. The learned similarity metric later allows the matching of new persons from faces not seen during training. The authors in [4] trained and tested this technique on input face images from a combination of the AR dataset and the FERET dataset. They achieved a verification equal error rate (EER) of 2.5%, though the network partially saw sub-

jects used for testing during training. This technique's strength is that invariant effects do not come from previous knowledge about the task, but are learned during training. This overcomes the shortcomings of previous techniques that are sensitive to geometric transformations and distortions in the input image pairs. However, due to the complex architecture of CNNs, this system is inefficient in terms of speed. The authors in [10] improved Chopra's design in terms of computational speed and complexity by fusing the convolutional and subsampling layers of the CNNs in the model, making it a four-layer CNN architecture (an idea introduced by [14] in handwriting digit recognition). Figure 3 shows the change in the architecture of the convolutional neural subnets. These authors applied this model to the AT&T dataset and achieved an equal error rate (EER) of 3.33%. This technique could classify a pair of images in 0.6 milliseconds, which is significantly faster than [4]. It could also verify test subjects not seen during training.

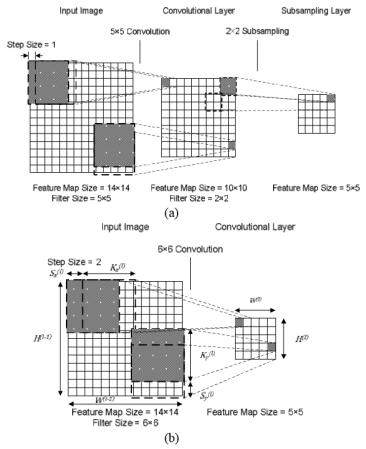

Figure 3: *Convolution layer (a) Convolution (with stride of 1) followed by subsampling; (b) convolution operation (with stride of 2) [10].*

There have been different variants of deep convolution networks that differ in model architecture through the past years. Some popular ones are explained in the papers [11] [13] [15] [18] [8] [7] [9]. Some say the paper [12] was the pioneering publication, but [11] is regarded as the most influential paper. The architecture of the network, called AlexNet, paved the way for CNNs. It has a relatively simple layout architecture, and it achieved a top 5 test error rate of 15.4% (top 5 error is the rate at which, given an image, the model does not output the correct label with its top 5 predictions). In 2012 the model ZF Net [13] fine-tuned AlexNet to improve GPUs' performance and achieved an 11.2% error rate. In this paper, the authors clearly show how to visualize the filters and weights correctly. Most past proposed models and techniques extract features from global facial images. None of them explore the viability of Patch-based feature extraction. In this paper, we will explore the viability of region-specific feature extraction with Siamese CNNs.

III. Methods

i. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) or ConvNets have had a significant advancement in image analysis due to their specialization characteristics by detecting patterns and making sense of them. The main three layers of a CNN that enable this specialization are: the Convolution layer, pooling/subsampling layer, and fully-connected layer. The network's convolution layer receives an input image and outputs a stack of filtered images (feature maps) to the next layer using the convolution operator. The number of output feature maps is determined by what we have set as the number of filters. This layer detects different features using different filters (edges, shapes, texture, objects, etc.). A filter is simply a small matrix, and we determine the dimensions. The values of the filter are initially randomized and are learned during training. The deeper the network goes, the more sophisticated the filter becomes such that rather than detecting edges or shapes, they may be able to detect specific objects like eyes and nose.

The general equation for spatial filtering (cor-

a	b	c
d	**e**	f
g	h	i

Figure 4: *Original image Pixels*

r	s	t
u	**v**	w
x	y	z

Figure 5: *Filter*

relation) is:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{t} w(s,t) f(x+s, y+t) \quad (1)$$

where x and y are varied so that each pixel in w visits every pixel in f. This can be expressed as equation 2 given the original image pixels (Figure 4) and filter (Figure 5) shown.

$$e_{(processed)} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i \quad (2)$$

Convolution works the same except the filter is first rotated by 180°. The general equation for the convolution operator is therefore given by:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{t} w(s,t) f(x-s, y-t) \quad (3)$$

This can be expressed as equation 4 given the original image pixels and filter shown above.

$$e_{(processed)} = v*e + z*a + y*b + x*c + w*d + u*f + t*g + s*h + r*i \quad (4)$$

The pooling/subsampling layer shrinks the stack of feature maps by downsampling the features, so that the model learns fewer parameters during training, reducing the chance of over-fitting. This is done by stepping through each of the filtered images from the convolution layer with a filter of a particular window (usually two) and by a particular stride (usually two). Equation 5 and 6 gives the width and height of the resulting images after pooling. Max pooling is a typical filtering operation used for this layer. It works by taking a maximum value from each window, and this works better than average pooling.

$$output_w = Image_w - Filter_w + 1 \quad (5)$$

$$output_h = Image_h - Filter_h + 1 \quad (6)$$

The pooling/subsampling layer usually includes an activation function or put as a separate layer. We will use the Rectified linear activation function (ReLu). The function steps through every pixel in a given image, returning it directly if it is positive. Otherwise, it will return zero as shown in Figure 6. Instead of the sigmoid or hyperbolic tangent activation function, we will use this function to avoid the vanishing gradient problem.

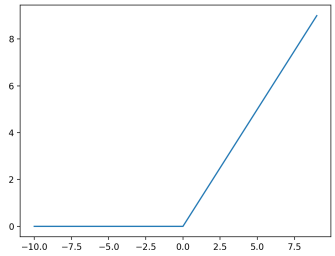

Figure 6: *Rectified Linear Activation for Negative and Positive Inputs [3].*

The last layer of the network is the fully-connected layer. This layer takes a list of flattened feature values from the last pooling layer into a one-directional feature vector. This architecture's rationale is that the convolution layer provides a low dimension, invariant feature space, and a fully connected layer learning a non-linear function in that space. The learning happens through back-propagation and gradient descent. The model learns features (filter

values) in the convolution layer and weights in the fully-connected layer.

ii. Siamese Architecture

The siamese architecture is a network of two identical neural networks that share weights. It receives two inputs and returns a similarity measure that tells us how similar the two inputs are. The two neural networks are replaced with convolutional neural networks and applied to facial recognition. The CNNs get us two feature vectors that give us a similarity measure by taking the element-wise absolute difference. Through this, we can deduce if they are genuine pairs or imposter pairs. A distance function is learned between the two vector representations produced by the same neural networks, such that two equal faces would have: similar feature vectors, a small absolute difference, and a high similarity score. In contrast, two different faces would have: different feature vectors, a high absolute difference, and a low similarity score. Equation 7 is used to compute the pair-wise distance between the two output vectors using the p-norm. During training, we propagate and update the model parameters so that the conditions above are satisfied.

$$\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}} \tag{7}$$

Where p is the norm degree.

iii. Loss Function

A loss function is used to calculate the model error. Its job is to represent all aspects of the model into a single number, and improvements on that number signify a better model. We use the pair-wise ranking loss which can be expressed as equation 8 or 9. The functions compare a query input image (q) against a reference input image (r). The query may be a genuine match (q_p) or an imposter (q_n). A margin (m) is used to create a minimum distance between positive and negative queries.

$$L(r_a, q, m) = \begin{cases} d(r_a, q) & if q = q_p \\ max(0, m - d(r_a, q)) & if q = q_n \end{cases} \tag{8}$$

$$L(r_a, q, y) = y\|r_a - q\| + (1 - y)max(0, m - \|r_a - q\|) \tag{9}$$

iv. Cascade Classifier Model

Patch detection and cascading are vital steps in this experiment. We train our own Haar cascade classifier to detect the eye region, illustrated in Figure 7, an approach adapted from [20]. Viola and Jones describes a machine learning approach to visual object detection that uses three techniques: Integral image, AdaBoost, and Cascading. The integral image is a representation of an image that allows any Haar-like feature used by the detector to be computed more efficiently. AdaBoost [6] allows the detector to focus on a smaller set of Haar-like features given a more extensive set. The third technique is combining increasingly complex classifiers in a cascade structure such that the detector focuses on promising regions of the image.

Figure 7: *Face and Eye Region Patch Detection.*

In creating our own Haar cascade, we use the Cascade Trainer GUI [1] that allows comfortable use of OpenCV tools for training cascade classifier models. We used the GUI in the following steps:

1. Collect a dataset of positive images (those containing the object to be detected) and negative images (those that do not contain the object to be detected) images.
2. Create a folder p of positive images.
3. Create a folder n of negative images.
4. Use the Haar Trainer GUI to train by specifying the path to the two folders, set the number of stages, and set the width and height of positive images.

For this research a classifier described above was trained on 200 positive samples of 62×22 window size, which gives about 905498 Haar features per window. One thousand negative samples were used. The training was done in 20 stages, which took about 8 hours.

v. Model architecture

The CNN structure below was used for all models in this research. All input images to each convolutional layer are padded using reflection padding of size 1. We make normalization a part of the model architecture by performing batch normalization on each output feature map to the next convolutional layer. C_x denotes a fused convolutional/subsampling layer. F_x denotes a fully connected layer that applies a linear transformation to input data.

- C1: Feature maps:5; kernel size 3×3; stride:2; input: 1×100×100 image; output feature maps 5×50×50.
- C2: Feature maps:14; kernel size 3×3; stride:2; input: 5×50×50 image; output feature maps 14×25×25.
- C3: Feature maps:60; kernel size 3×3; stride:2; input: 14×25×25 image; output feature maps 60×13×13.
- F4: input 10140 features; output 3200 features.
- F5: input 3200 features; output 1600 features.
- F6: input 1600 features; output 40 features.

A siamese CNN, Figure 8, is then two of the CNN subnets described above where each outputs 40 features describing an input image. Equation 7 is used to compute the pair-wise distance between the two output vectors using the p-norm. Equation 8 or 9 gives us the pair-wise ranking loss. Weight updates are back-propagated using the Adaptive Momentum Optimization Algorithm (Adam), with a learning rate of 0.0005, which is an optimization of both the Stochastic Gradient + momentum (SGD + momentum) and the Root Mean Squared Propagation (RMSProp).

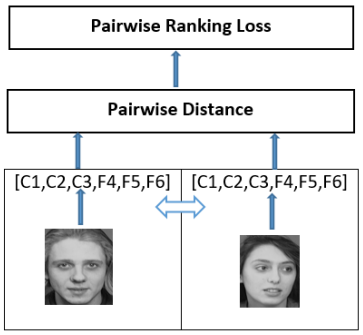

Figure 8: *Siamese CNN Architecture with an Imposter Input Pair.*

IV. Experimental methodology

i. Dataset

The model described above was trained and tested on the AT&T dataset [5]. The dataset consists of 40 different subjects, each with ten different gray-scale facial images. These images are taken against a dark homogeneous background with subjects in an upright and frontal position (with tolerance for some side movement). The images vary in facial expressions (open eyes/closed eyes/smiling/not smiling) and facial details (glasses/ no glasses), and lighting. The dataset was partitioned into two disjoint sets: the training set with thirty five individuals and the testing set with five individuals. Though this research assumes accurate detection, detection is, however, still a required step. Therefore, the dataset is preprocessed to remove challenging cases in which

the face or eye region could not be detected, such that only 274 were used for training and 42 for testing. From these images, genuine pairs (images of the same person) and imposter pairs (images of different people) were created—37401 for training and 861 for testing.

ii. Data Pre-processing

Eye region detection and patch extraction performed for the Patch-based CNN prior to feeding each patch to the respective model. The eye region is detected using the method described above. The positions of the rest of the patches are deduced and extracted from there. Nine patches were extracted, Figure 9:

- Patch 1: Eye region
- Patch 2: Eye region + forehead
- Patch 3: Nose
- Patch 4: Eye region and nose
- Patch 5: Jaw region and mouth
- Patch 6: From eye region to the chin
- Patch 7: Nose and cheeks to chin.
- Patch 8: Left half of the face.
- Patch 9: Right half of the face.

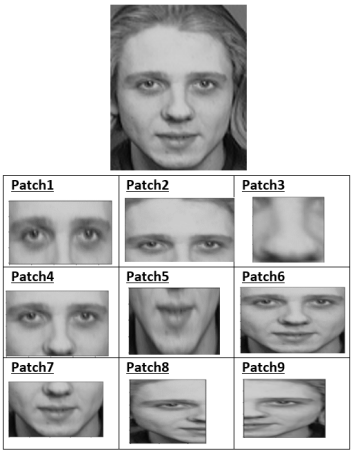

Figure 9: *Example of 9 Patches extracted from a single input image.*

iii. Global Siamese CNN

The global Siamese CNN is shown in Figure 10. The CNN's output is a dissimilarity measure of the two input images: high for imposter pairs and close to 0 for genuine pairs. By comparing the dissimilarity measure to a given threshold, the model predicts the pair as genuine or imposter.

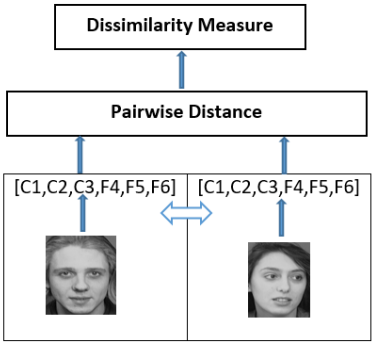

Figure 10: *Global Siamese CNN Architecture.*

iv. Patch-based Siamese CNN

The Patch-based CNN is a set of Siamese CNNs where each Siamese CNN specializes in comparing a specific patch described in the sections above. We trained 9 Siamese CNNs. Any combination of Patch-specific CNN would make up a Patch-based model, as illustrated in Figure 11.

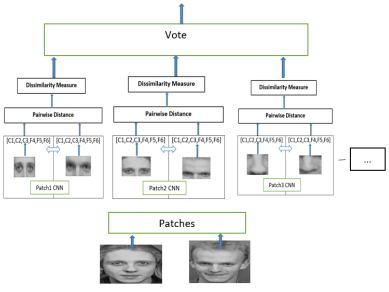

Figure 11: *Patch-Based Siamese CNN Architecture.*

Each model in the combination outputs a dissimilarity measure. We aggregate them into an ensemble by a voting mechanism. The combination predicts a given pair to be genuine if

most of the Patch-specific models are predicted as genuine, given some threshold. Otherwise, the pair is predicted as an imposter pair.

V. Results

We test the effectiveness of the proposed technique through accuracy. Accuracy is deduced from the equal error rate (EER) derived from a combination of false acceptance rate (FAR) and false rejection rate (FRR) on our test dataset. FAR is the likelihood of the model incorrectly accepting an imposter sample, equation 10. FRR is the likelihood of the model incorrectly rejecting a genuine sample, equation 11. EER is the value when FAR is equal to FRR. The lower the EER, the higher the accuracy of the model.

$$FAR = \frac{FA}{FA + TN} \tag{10}$$

$$FRR = 1 - TPR \tag{11}$$

Where

$$TPR = \frac{TP}{TP + FR} \tag{12}$$

FA is the False acceptance count, TP is the True Positive count, TN is the True Negative count and FR is the False Rejection count.

i. Global Siamese CNN

Results for the global CNN on the dataset described above are shown in Figures 12, 13 and 14. The model achieved an EER of 0.090 at a threshold of 0.27.

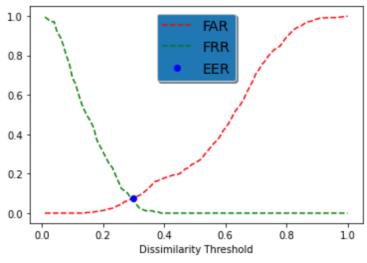

Figure 12: *Global CNN: FAR, FRR and EER.*

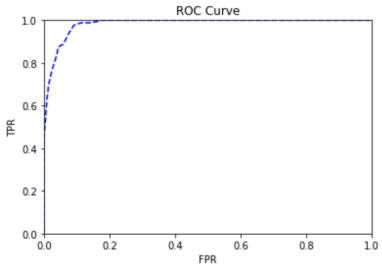

Figure 13: *Global CNN: ROC Curve.*

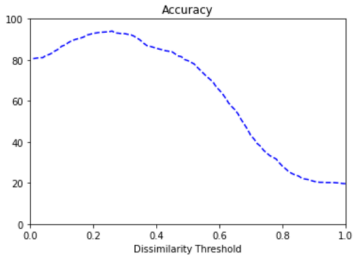

Figure 14: *Global CNN: Accuracy Per Threshold.*

ii. Patch-Specific CNNs

To create accurate combinations of Patch-based CNNs, we first evaluated the performance of each CNN in isolation. Results are shown in the table below. From the nine patches extracted, one could create many combinations; we evaluated three combinations:

- Combination 1: Made of patches 1, 2, 3, 4, and 5.
- Combination 2: Made of patches 4, 5, 6, 8, and 9.
- Combination 3: Made of all the patches.

Table 1: *EER And Threshold Of Each Patch-specific model*

Patch	EER	Threshold
Patch1	0.247	0.29
Patch2	0.30	0.32
Patch3	0.29	0.36
Patch4	0.074	0.23
Patch5	0.173	0.37
Patch6	0.063	0.29
Patch7	0.167	0.34
Patch8	0.079	0.23
Patch9	0.166	0.35

iii. Combination 1

Combination 1 achieved an EER of 0.098 at a threshold of 0.33. Results are shown in Figures 15, 16 and 17.

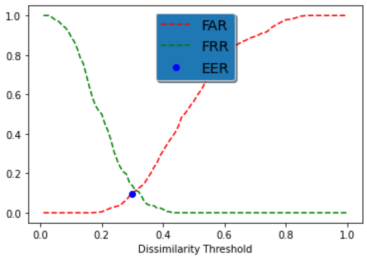

Figure 15: *Patch-based CNN of the first combination of patches: FAR, FRR and EER.*

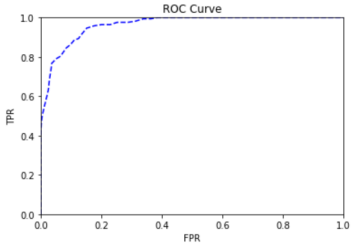

Figure 16: *Patch-based CNN of the first combination of patches: ROC Curve*

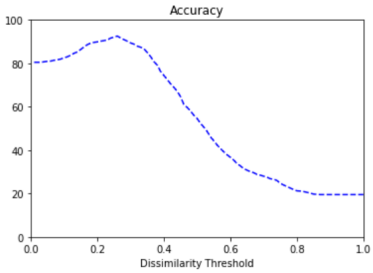

Figure 17: *Patch-based CNN of the first combination of patches: Accuracy Per Threshold*

iv. Combination 2

Combination 2 achieved an EER of 0.037 at a threshold of 0.35. Results are shown in Figures 18, 19 and 20.

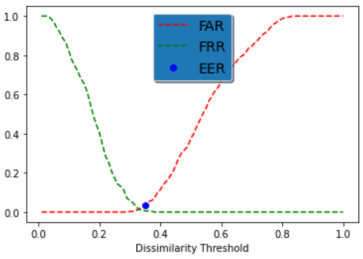

Figure 18: *Patch-based CNN of the second combination of patches: FAR, FRR and EER.*

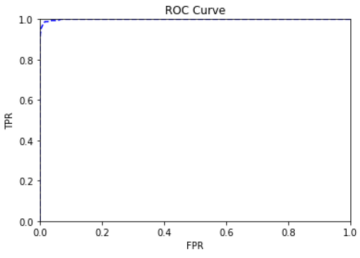

Figure 19: *Patch-based CNN of the second combination of patches: ROC Curve*

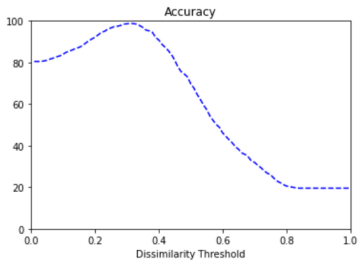

Figure 20: *Patch-based CNN of the second combination of patches: Accuracy Per Threshold*

v. Combination 3

Combination 3 achieved an EER of 0.045 at a threshold of 0.32. Results are shown in Figures 21, 22 and 23

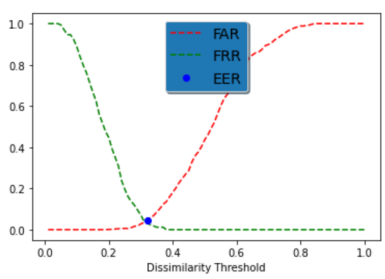

Figure 21: *Patch-based CNN of the third combination of patches: FAR, FRR and EER.*

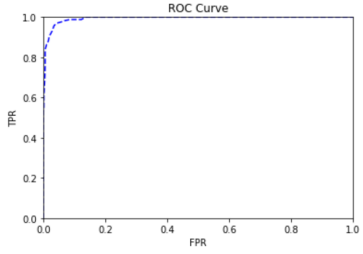

Figure 22: *Patch-based CNN of the third combination of patches: ROC Curve*

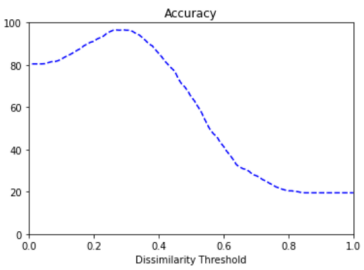

Figure 23: *Patch-based CNN of the third combination of patches: Accuracy Per Threshold*

VI. Discussion And Conclusion

This paper describes a new insight that some specific patches are more effective in accurate facial recognition than others (Table 1). Detailed experiments presented in this paper confirm two facts.

Firstly, we could use more facial information as patches instead of one global face and achieve better accuracy. This is preferred when there is a possibility of presenting a partially deformed face. Since the combination (a Patch-based Siamese CNN) will predict according to the majority of its Patch-specific Siamese sub-models, it will remain discriminative enough in such situations. Further experiments show that there is a task in choosing the correct combination of Patch-specific sub-models. If a correct combination, like combination 2, that combines sub-models of a low EER is chosen, the results are favorable. Compared to the results of a random combination, like combination 1. The dissimilarity threshold at the EER position of a Patch-based CNN is more than that of a global CNN suggests that the Patch-based CNN is more lenient. A higher dissimilarity threshold means that the model maintains high accuracy even though it allows a greater difference between the two input images and still predicts them as a genuine pair.

Secondly, table 1 shows that if a correct patch is chosen (like patch 4), one patch that is less than the entire face can be more accurate

than presenting a global face. We could create a facial recognition system that uses less facial information, therefore, decreasing computational demand.

We can conclude that this study answers two questions: A Patch-based feature extraction can be more effective than global feature extraction. Less information (a partial image) can be used instead of the entire face and maintain accuracy.

VII. Future Works

The performance of the proposed technique depends on accurate detection that leads to patch extraction. Exploring various detection methods and pre-processing methods may increase the accuracy of a Patch-based CNN. Accuracy also depends on combining the right set of Patch-specific sub-models to make up a Patch-based model. A possible extension to this research is adding a method to dynamically choose and combine patches depending on the situation presented. The more patches extracted, the greater the computational demand. Parallel computing can be explored such that Patch-specific models run in parallel and present a decision to a central component for voting.

References

[1] Amin Ahmadi. *Cascade trainer GUI*. 2016.

[2] Jane Bromley et al. "Signature Verification using a Siamese Time Delay Neural Network". In: *Int.]. Pattern Recognit. Artzf Intell* 7 (1993).

[3] Jason Brownlee. "A gentle introduction to the rectified linear unit (relu)". In: *Machine Learning Mastery. https://machinelearningmastery.com/rectified-linear-activation-functionfordeep-learning-neural-networks* (2019).

[4] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 539–546.

[5] Andy Hopper FREng. "The ORL face database". In: *AT&T (Olivetti) Research Laboratory Cambridge,"ht tp://www. uk. research. att. com/facedatabase. html* 199.2 (1992).

[6] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of online learning and an application to boosting". In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[7] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[8] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[9] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. "Spatial transformer networks". In: *Advances in neural information processing systems*. 2015, pp. 2017–2025.

[10] Mohamed Khalil-Hani and Liew Shan Sung. "A convolutional neural network approach for face verification". In: *2014 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE. 2014, pp. 707–714.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[12] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[13] D Matthew and R Fergus. "Visualizing and understanding convolutional neural networks". In: *Proceedings of the 13th European Conference Computer Vision and Pattern Recognition, Zurich, Switzerland*. 2014, pp. 6–12.

[14] Patrice Y Simard, David Steinkraus, John C Platt, et al. "Best practices for convolutional neural networks applied to visual document analysis." In: *Icdar*. Vol. 3. 2003. 2003.

[15] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[16] William Stallings et al. *Computer security: principles and practice*. Pearson Education Upper Saddle River, NJ, USA, 2012.

[17] Michal Strahilevitz. "Facial Recognition Bans: What Do They Mean For AI (Artificial Intelligence)?" In: (2020).

[18] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[19] Matthew Turk and Alex Pentland. "Eigenfaces for recognition". In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86.

[20] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. IEEE. 2001, pp. I–I.