Face recognition using deep learning

1st Ziad Tarek
Computer engineering
Nile University
211001879
6th Ocotber, Egypt
z.tarek2179@nu.edu.eg

2nd Youssef Emad
Communication Engineer
Nile University
202000608
6th October,Egypt
yo.emad@nu.edu.eg

3th Omar Ayman
Computer Engineer
Nile University
202000537
6th October,Egypt
o.ayman@nu.edu.eg

4th Mohamed Yasser
Computer Engineer
Nile University
202000430
6th October,Egypt
m.ghandour@nu.edu.eg

5th Zeyad Khaled
Communication Engineer
Nile University
202001042
6th October,Egypt
z.abdelaziz@nu.edu.eg

***Abstract*—In this paper, we will make a face verification application using neural networks which use a special structure to rank similarity among inputs, The industry needs to create effective and automated facial recognition systems for a number of new applications. Face recognition is the process of analysing the features of a person's face photos that are captured online or by a digital video camera. For face recognition, several algorithms have been propsed; many are currently being developed. This review paper discusses the many neural network methods that have been put out by numerous researchers for use in facial recognition systems. Since the early 1980s, there has been an increase in interest in the study of neural networks. These methods don't require a precise mathematical model of the process, merely relevant training data.In this paper we will make this real life application that uses neural network to make verifications that it is the same user in the camera, we will achieve that by taking an image of the user and we will generate based on it around 1500 other varieties by changing contrast ,flipping,changing saturation or adding noise to it,doing this will achieve higher accuracy, we will use jupyter notebook and python opencv library for this application.**

I. Introduction

The human face is a crucial aspect of social communication and interaction. Humans need to recognize the face of others for these purposes. Throughout his whole life, a person has to recognize thousands of other persons' faces surrounding him. For human-computer interaction, face recognition is also essential. Nowadays, it is also widely used in access control, security, surveillance systems, the entertainment industry. [1]

Due to changes in facial expressions, positions, and lighting, face recognition is an extremely difficult research subject in computer vision and pattern recognition. The industry must create effective and automated facial recognition systems for a number of new applications, including commercial and law enforcement jobs. Even though numerous academics have been working on the issue of facial recognition for many years, there are still a number of problems that need to be resolved. Changes in pose, position, and expression are just a few examples of some of the challenges that need to be handled carefully. Additionally, as the quantity of the face database grows, the recognition time becomes a significant obstacle. One biometric technique that has the advantages of high accuracy and minimal intrusion is face recognition.It is accurate yet not intrusive, like a physiological approach. For this reason, academics in a variety of disciplines including security, psychology, image processing, and computer vision have been interested in face recognition. For face recognition, many different algorithms have been proposed. [2]

A subfield of machine learning is deep learning. Deep learning is more suited for handling large amounts of data than regular machine learning. The performance of algorithms improves as data volume grows. Deep learning does not rely on the artificial determination of application properties like typical machine learning does. Instead, it makes several feature transformations in order to acquire higher-level features straight from the input and create a deep machine learning model. [3]

Applications can be categorised into two classes, known as single sample per person (SSPP) and multiple samples per person (MSPP) problems, depending on whether one or more samples are available during training or when users are enrolled in the system. Due to the importance of its applications, particularly those relating to security, surveillance, and border crossing, the SSPP problem has attracted a lot of interest during the past few decades. These days, e-voting and banks of the future service providers are becoming more and more popular. These apps must ensure transaction security while simultaneously providing a pleasing, practical, and comfortable user experience, including mobile device access,Consequently, they normally avoid storing any user information and instead compare a user's photo to the photo on his or her ID card. Lower sample collection, storage, and processing costs are offered by SSPP scenarios, but they also bring new obstacles to the field of face recognition. Since most current methods heavily rely on the quantity and quality of samples to create a good facial model that generalises both inter- and intra-person variability, the drop in the number of photos means a serious reduction in the recognition accuracy. Therefore, a solution that can draw information from prior experiences and surroundings like these and generalise it is needed. [4] In general, siamese neural networks are used to learn picture representations via a supervised metric-based method, and then the features from that network are reused for one-shot learning without any retraining.

In general, siamese neural networks are used to learn picture representations via a supervised metric-based method, and then the features from that network are reused for one-shot learning without any retraining. Character recognition is the focus of our investigations, but the fundamental methodology can be applied to nearly any modality. Large siamese convolutional neural networks are used in this domain because they can: a) learn generic image features useful for predicting unknown class distributions even when there are few examples available; b) easily train using standard optimization techniques on pairs sampled from the source data; and c) provide a targeted approach that does not rely on domain-specific knowledge by instead utilising [**?**]

This paper will discuss face recognition's early algorithms, artificial features and classifiers, deep learning, and other stages of development.After that, we will introduce the research on face recognition for real conditions. Finally, we will make a face recognition application using the datasets ,the machine learning libraries and python library opencv.

II. Literature Review

Prior to the development of deep learning algorithms, the majority of existing face recognition techniques relied on hand-crafted shallow local features extracted from facial images using Local Binary Patterns (LBP), Scale Invariant Feature Transform (SIFT), and Histogram of Oriented Gradients (HOG), followed by feature training and identity classification using Nearest Neighbors (NNs) or Support Vector Machines. However, due to the availability of state-of-the-art computational capabilities and increased access to very large datasets, deep learning architectures have been developed and have produced incredibly excellent results for a variety of visual recognition tasks, including face recognition. [5]

Simple statistical techniques were used to begin face detection by a machine. One of the most well-known of them was Eigenfaces. Every picture is represented as a vector of weights that were produced by projecting on eigenface components For face recognition, researchers have also attempted to apply several other established techniques, such as elastic graph matching, Karhunen-Loeve based algorithms [4], and singular value decomposition. Most of the time, modest datasets were used to evaluate these strategies. The dataset size was even fewer than 100 in several instances. Although statistical methods are not entirely effective, they provide assurance that the system itself can recognise the human face without outside manipulation. It has a permanent, positive effect on future development. [1] [4]

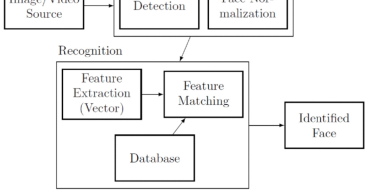


Fig. 1. The algorithm behind face recognition.

The ground-breaking research on one-shot learning was done by Li Fei-Fei et al. in the early 2000s. The authors created a variational Bayesian framework for one-shot picture classification on the basis that learned classes can be used to predict new ones when there aren't enough examples of a specific class available (Fe-Fei et al., 2003; Fei-Fei et al., 2006). More recently, Lake et al. used a technique called Hierarchical Bayesian Program Learning (HBPL) to solve one-shot learning for character recognition. They did this by approaching the problem of one-shot learning from the perspective of cognitive science (2013). The process of generating character drawings was modelled by the authors in a number of works in order to fragment the final image (Lake et al., 2011; 2012).Finding a structural explanation for the observed pixels is the aim of HBPL. However, because the combined parameter space is so big and presents an insoluble integration problem, inference under HBPL is challenging.

In paper [5] it presents and describes two successful CNN architectures for face recognition and discuss face representation based on these models. First VGG-Face Network A deep convolutional network called VGG-Facial has been suggested using the VGGNet architecture for face recognition. On 2.6 million facial photos representing 2,622 identities gathered from the internet, it was trained. 16 convolutional layers, 5 max-pooling layers, 3 fully linked layers, and a final linear layer with Softmax activation make up the network. Dropout regularisation [28] is used in the fully-connected layers of VGG-Face, which uses colour image patches of size 224 224 pixels as input. Additionally, it activates all of its convolutional layers using ReLU. It is abundantly obvious from 144 million parameters that the VGG network is a computationally costly architecture. 35 On the LFW dataset, this method was tested, and the accuracy was 98.95

Secondly Lightened CNN.This face recognition framework is a CNN with a low level of computational complexity. In compared to the ReLU, it extracts more abstract representations using an activation function called Max-Feature-Map (MFM). In two distinct models, lighter CNN is introduced. The first network (A), which was modelled after the AlexNet model, has 3,961K parameters, four convolutional layers that use MFM activation functions, four max-pooling layers, two fully connected layers, and an output linear layer that uses Softmax activation. The second network (B), which is modelled after the Network in Network theory, has 3,244K parameters, five convolutional layers that use MFM activation functions, four convolutional layers for dimensionality reduction, five max-pooling layers, two fully connected layers, and a linear layer with Softmax activation in the output.Grayscale facial patch photographs with a size of 128128 pixels are used as network inputs by the Lightened CNN models. The CASIA WebFace dataset contains 493,456 facial photographs representing 10,575 identities and is used to train these models. On the LFW dataset, both Lightened CNN models were assessed, and their respective accuracies were 98.13 and 97.77. Pre-trained models of VGG-Face and Lightened CNN are used in the Caffe deep learning framework. The first fully-connected layer with Softmax activation is indicated as FC6 and FC1. To analyze the effects of differentfully-connected layers, we also deploy the FC7 layer of the VGA-Face network. [5]

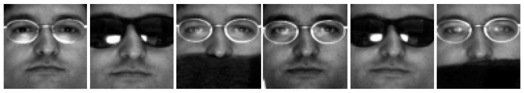


Fig. 2. Samples from the AR database with different occlusion conditions. The first three images from left are associated with Session 1 and the next three are obtained from Session 2 with repeating conditions of neutral, wearing a pair of sunglasses, and wearing a scarf.

Results for this experiment

Testing Set	VGG-Face FC6 FC7	Lightened
Sunglasses Session	1 33.64 35.45	5.45 (A)
Scarf Session 1	86.36 89.09	12.73 (A)
Sunglasses Session 2	29.09 28.18	7.27 (B)
Scarf Session 2	85.45 83.64	10.00 (A)

table1

As seen in Table 1, deep face representation struggles to cope with upper face occlusion brought on by sunglasses. The obtained results using deep representation are somewhat poor when compared to the most advanced occlusion-robust face recognition algorithms . These findings suggest that deep CNN-based representation may not perform well in the presence of facial occlusion unless specially trained on a huge amount of data with occlusion. The VGG-Face model is discovered in the same studies to be more resistant to face occlusion than the Lightened CNN models. Only the outcomes of the top-performing Lightened CNN models are shown in this table.

To summarize it has been shown that deep learning based representations provide promising results. However, the achieved performance levels are not as high as those from the state-of the-art methods The performance gap is significant for the cases in which the tested conditions are scarce in the training datasets of CNN models

III. Methodology

Our standard model is a siamese convolutional neural network with L layers of Nl units each, where h1,l represents the hidden vector in layer l for the first twin and h2,l represents the same for the second twin. In the first L 2 layers, we only employ rectified linear (ReLU) units.

The remaining levels contain sigmoidal units. The model is made up of a series of convolutional layers, each of which employs a single channel with varying size filters and a fixed stride of 1. To optimise efficiency, the number of convolutional filters is specified as a multiple of 16. The network activates the output feature maps with a ReLU activation function, which is optionally followed by maxpooling with a filter size and stride of 2 Thus the kth filter map in each layer takes the following form:

$$a(k)_{1,m} = max - pool(max(0, W_{(k)l}1, l). * h1, (l1) + bl, 2) \tag{1}$$

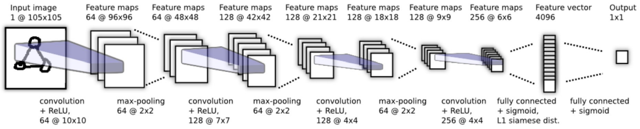


Fig. 3. The best convolutional architecture was chosen for the verification task. The Siamese twin is not shown, although it joins immediately after the 4096 unit fully-connected layer, which computes the L1 component-wise distance between vectors.

$$a(k)_{2,m} = max - pool(max(0, W(k)l1, l. * h2, (l1) + bl), 2) \tag{2}$$

where Wl1,l is the 3-dimensional tensor representing the feature mappings for layer L and .* to be the correct convolutional operation for returning

Only the output units resulting from complete overlap between each convolutional filter and the input feature maps were considered.

The last convolutional layer's units are flattened into a single vector. This convolutional layer is followed by a fully connected layer, and then another layer that computes the induced distance metric between each siamese twin and outputs it to a single sigmoidal output unit. To be more exact, the prediction vector is given as

$$p = \alpha(\sum_j \alpha_j |h_(j)1, L1h_(j)2, L1|) \tag{3}$$

where alpha is the sigmoidal activation function. This final layer applies a metric to the learnt feature space of the (L 1)th hidden layer and ranks the similarity of the two feature vectors. The j are extra parameters acquired by the model during training that weight the importance of the component-wise distance. This establishes the network's last Lth completely linked layer, which connects the two siamese twins. Figure 3 displays example of our model that we investigated. This network also performed the best of any network on the verification task.

Optimization: Due to the connected weights, the gradient is cumulative across the twin networks when this objective is paired with a regular backpropagation technique. We decide on a minibatch size of 128 with a learning rate of j and momentum.j, and L2 regularisation weights j defined layer-wise, resulting in the following update rule at epoch T:

$$w(T)kj(x(i)1, x(i)2) = w(T)kj + \delta w_j(T)k(x(i)1, x(i)2) + 2\lambda jWkj| \tag{4}$$

$$\alpha w(T)kj(x(i)1, x(i)2) = \eta_j w(T)kj + \mu_j \alpha w_j(T1)k \tag{5}$$

where $\vec{\Delta}$ wkj is the partial derivative with respect to the weight between the jth neuron in some layer and the kth neuron in the successive layer.

Learning schedule: learning rates were decayed uniformly across the network by 1 percent per epoch, so that

$$\eta_j(T) = 0.99\eta_j(T1) \tag{6}$$

. We found that by annealing the learning rate, the network was able to converge to local minima more easily without getting stuck in the error surface. We trained each network for a maximum of 200 epochs while measuring one-shot validation error on a collection of 300 oneshot learning images created at random from the lfw data set. When the validation error did not reduce after 11 epochs, we terminated and applied the model parameters at the best epoch based on the one-shot validation error. We saved the model's final state as a result of this approach.

Practical use for the model: We used lfw dataset which is a library containing over 5000 photos of random people which helps in making the model more accurate after training. for the negatives and used self images for anchors and positives, lfw dataset is a library containing over 5000 photos of random people which helps in making the model more accurate after training.

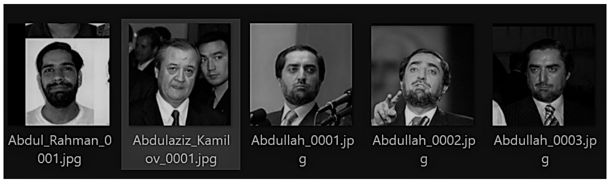


Fig. 4. Random faces for training the model to determine the negatives

for testing positives we took a sample of 300 images of ourselves and then made a copy of each image with a change in contrast, brightness, flipping and sharpness to get accurate results under any lightning, then we used them to train the model to detrmine the positives by taking the anchor image comparing it with the positive images then comparing it a again with negative images to make sure it differentiates between positives and negatives.

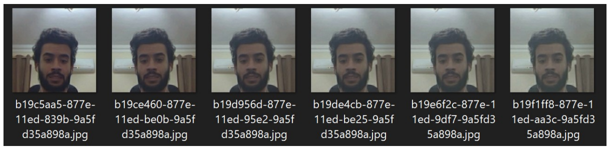

b19c5aa5-877e-11ed-839b-9a5fd35a898a.jpg
b19ce460-877e-11ed-be0b-9a5fd35a898a.jpg
b19d956d-877e-11ed-95e2-9a5fd35a898a.jpg
b19de4cb-877e-11ed-be25-9a5fd35a898a.jpg
b19e6f2c-877e-11ed-9df7-9a5fd35a898a.jpg
b19f1ff8-877e-11ed-aa3c-9a5fd35a898a.jpg

Fig. 5. Same photo flipped,changed in contrast , brightness, sharpness

The last step was build our model in an application we have done that by using kivy, a python library , we imported our distance layer ,similarity calculation and the trained model and then built a ui to facilitate the use of it.

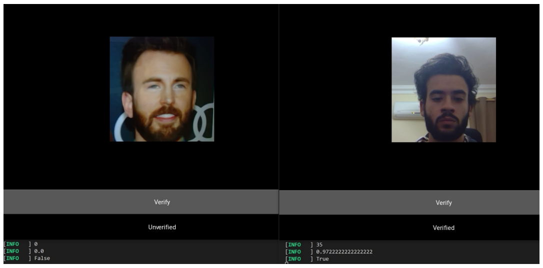
IV. Results

We are now ready to demonstrate the discriminative capability of our learned features at one-shot learning after optimising a siamese network to master the verification task. Except for HBPL, our convolutional method outperforms all others at 92 percent. This is just somewhat lower than the rate of human error. While HBPL produces better overall outcomes, our top-performing convolutional network did not.

Table 2: Comparing best one-shot accuracy from each type of network against baselines.

Method	Test
Humans	95.5
Hierarchical Bayesian Program Learning	95.2
Affine model	81.8
Hierarchical Deep	65.2
Deep Boltzmann Machine	62.0
Simple Stroke	35.2
1-Nearest Neighbor	21.7
Siamese Neural Net	58.3
Convolutional Siamese Net	92.0

after trainimg the model we wanted to put it in a practical use so we used kivy which is Kivy is a free and open source Python framework for developing desktop and mobile apps.



```
[INFO   ] 0
[INFO   ] 0.0
[INFO   ] False
```

```
[INFO   ] 35
[INFO   ] 0.9722222222222222
[INFO   ] True
```

Fig. 6. Trying random person's photo on the app and then trying an image of the person which the model is trained to be positive on

As you can see in the first result on the left we got unverified with 0 images resulted positives in the comparison which means 100 % accuracy on this one, The second result on the left we got verifed with 35 images resulted positive with 97 % accuracy as the app is checking with total of 37 images.

V. Conclusion

We presented a method for doing one-shot classification that involves first learning deep convolutional siamese neural networks for verification. We presented fresh findings by comparing the performance of our networks to that of existing classifiers.Our networks surpass all known baselines by a wide margin and come close to the best results obtained by the previous authors.The main idea of the model is that it converts each image into layers and computes the distance layer then compares each image layer with the other and decide whether it is positive or not depending on the threshold that we decided.

We then used this model and used it an app that does a real-time verification with high accuracy. We hope that we achieve more accurate results that tends to 1 by training the model again with more images and by more different image qualities.

References

[1] M. T. H. Fuad, A. A. Fime, D. Sikder, M. A. R. Iftee, J. Rabbi, M. S. Al-Rakhami, A. Gumaei, O. Sen, M. Fuad, and M. N. Islam, "Recent advances in deep learning techniques for face recognition," *IEEE Access*, vol. 9, pp. 99112–99142, 2021.

[2] M. Kasar, D. Bhattacharyya, and T.-h. Kim, “Face recognition using neural network: A review,” *International Journal of Security and Its Applications*, vol. 10, pp. 81–100, 03 2016.

[3] Y.-n. Dong and G.-s. Liang, “Research and discussion on image recognition and classification algorithm based on deep learning,” in *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pp. 274–278, 2019.

[4] B. Ríos-Sánchez, D. Costa-da Silva, N. Martín-Yuste, and C. Sánchez-Ávila, “Deep learning for facial recognition on single sample per person scenarios with varied capturing conditions,” *Applied Sciences*, vol. 9, no. 24, 2019.

[5] M. Mehdipour Ghazi and H. Kemal Ekenel, “A comprehensive analysis of deep learning based representation for face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 34–41, 2016.