International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 6 Issue 1, November-December 2021 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

# VGGFace Transfer Learning and Siamese Network for Face Recognition

**Aafaq Altaf[1], Rahee Jan[1], Andleeb Qadir[1], Shahid Mohid-ud-din Bhat[2]**

[1]B Tech Student, Department of Computer Science Engineering,
[2]Assistant Professor, Department of Computer Science Engineering,
[1,2]SSM College of Engineering, Jammu and Kashmir, India

## ABSTRACT

Traditionally, data mining algorithms and machine learning algorithms are engineered to approach the problems in isolation. These algorithms are employed to train the model in separation on a specific feature space and same distribution. Depending on the business case, a model is trained by applying a machine learning algorithm for a specific task. A widespread assumption in the field of machine learning is that training data and test data must have identical feature spaces with the underlying distribution. On the contrary, in real world this assumption may not hold and thus models need to be rebuilt from the scratch if features and distribution changes. It is an arduous process to collect related training data and rebuild the models. In such cases, Transferring of Knowledge or transfer learning from disparate domains would be desirable. Transfer learning is a method of reusing a pre-trained model knowledge for another task. Transfer learning can be used for classification, regression and clustering problems. This paper uses one of the pre-trained models – VGGFace with Deep Convolutional Neural Network to classify images.

***KEYWORDS:*** *Deep Convolutional Neural Network, Image classification, Machine learning, Transfer learning, Siamese network, kaggle, VGG – 16*

***How to cite this paper:*** Aafaq Altaf | Rahee Jan | Andleeb Qadir | Shahid Mohid-ud-din Bhat "VGGFace Transfer Learning and Siamese Network for Face Recognition" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-6 | Issue-1, December 2021, pp.611-616, URL: www.ijtsrd.com/papers/ijtsrd47869.pdf

IJTSRD47869



## 1. INTRODUCTION

A **Facial Recognition System** is a technology that can capture a human face anywhere in an image or a video and also can find out its identity. A Face Recognition system has proven to be very beneficial in case of user identity verification in the recent past replacing the age-old authentication mechanisms like password protection or the One Time Passwords. In the last decade or so, we have seen a huge growth in the smart mobile industry for using face verification, and also numerous apps like Snapchat or Instagram which can put interesting filters on face. The ongoing research in this field has come up with a lot of scope in a wide range of applications, such as surveillance systems or law enforcement.

This paper is about exploring a popular algorithm running behind a Face Recognition system. Specifically, we are going to discuss the **Siamese Networks.** The Paper assumes that the reader has a basic understanding of Machine Learning and Deep Learning technologies and has some idea in building Neural Networks with Tensor Flow.

## 2. BACKGROUND

The term originally comes from the conjoined twin brothers Chang and Eng Bunker (May 11, 1811 — January 17, 1874), who were the first pair to be known internationally. The term is used for those twins who are physically connected to each other at the chest, or at the abdomen or the pelvis. The two individuals were originally from Thailand, formerly known as Siam, hence the name. The Neural Network we are going to see in this article also consists of a pair of Networks which are actually the same, hence the name derives from the Siamese Twins.

## 3. PROPOSED METHODOLOGY

As we saw above, the Siamese twins are connected physically, the Siamese network also consists of a pair of Neural Networks which are identical to each other, also known as **Sister Networks** Unlike a

conventional CNN, the **Siamese Network does not classify the images into certain categories or labels, rather it only finds out the distance between any two given images**. If the images have the same label, then the network should learn the parameters, i.e. the weights and the biases in such a way that it should produce a smaller distance between the two images, and if they belong to different labels, then the distance should be larger.

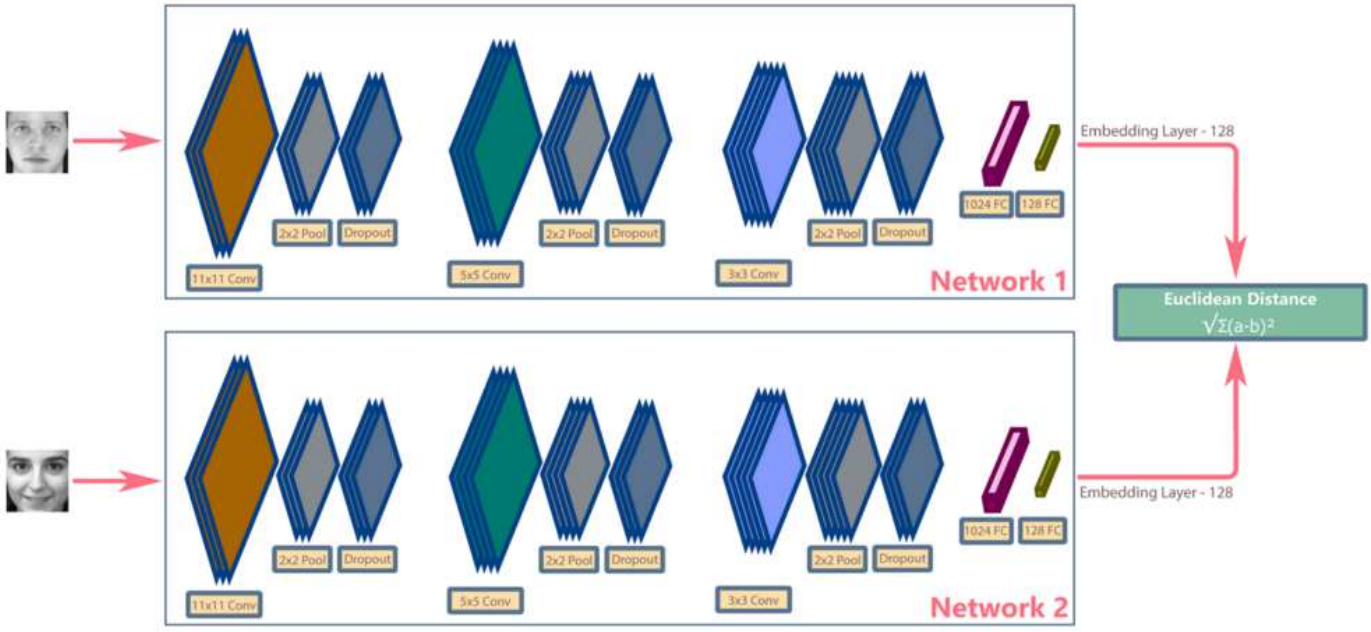


**Fig 1 Architecture of a Siamese Network**

As it shows in the diagram, the pair of the networks are the same. The Siamese Network works as follows.

To train a Siamese Network, **a pair of images** are picked from the dataset, each one processed by one of the networks above. (In next few sections, we will see how to generate pairs of images from the dataset.)

The networks have the same structure, hence the same operations will be performed on the respective images.

The Neural Networks at the end have **Fully Connected Layers**, with the last one consisting of 128 nodes. This layer is the final feature that gets produced when the network is applied on the image. It's called the **Embedding Layer Representation**. So the two images in the pair processed by the Siamese Network produce two different Embedding Layer Representations.

The Network then finds the **Euclidean distance** between both the embedding layers. If the images are of the same person, then it is expected that the embeddings will be very similar, hence distance should be smaller. However, if the images are of different people, then the distance is expected to be a higher value. A **Sigmoid Function** is applied on the distance value to bring it to 0–1 range.

A loss function is put on the sigmoid result, to penalize the network to update its weights and the biases. We are using **Binary Cross Entropy** in this paper for the loss function. Updation of the weights and the biases done on both the networks are exactly the same.

This process repeats for all the image pairs generated from the dataset. The same approach described above is put into code below. We have used Tensor Flow APIs to build the network architecture.

## 4. RESEARCH METHODS

```
importnumpyasnp
importmatplotlib.pyplotasplt
importcv2ascv
importos
importsys
fromtensorflow.keras.layersimportInput,Dense,Activation
fromtensorflow.keras.layersimportFlatten
fromtensorflow.keras.modelsimportModel
```

```
importtensorflowastf
fromtensorflow.keras.lossesimportBinaryCrossentropy
fromtensorflow.kerasimportbackendask
fromtensorflow.keras.optimizersimportAdam
fromtensorflow.keras.layersimportActivation
fromdeepface.basemodelsimportVGGFace
fromdeepface.basemodelsimportFacenet,VGGFace
vgg=VGGFace.baseModel()
vgg.load_weights('../input/weightsss/vgg_face_weights.h5')
https://www.kaggle.com/raafaq/weightsss/download
size=224
        shape=(size,size,3)
cnt=0
        forlyerinvgg.layers:
        ifcnt>34:
        lyer.trainable=True
        lyer.activation=tf.keras.layers.Activation('tanh')
        else:
        lyer.trainable=False
        cnt=cnt+1
```

*"By applying tanh activation function from layer 34, we can later use multiple images of single person for generating features and these features will be different . So it will be really helpful in face verification."*

```
        vgg_face=Model(inputs=vgg.layers[0].input,outputs=vgg.layers[-2].output)
        defcreate_model():
        inputs=Input(shape)
        #inputs = BatchNormalization(axis= -1)(inputs)
        outputs=vgg_face(inputs)
        x=Flatten()(outputs)

        model=Model(inputs,x)
        print(vgg_face.summary())
        returnmodel
extractor=create_model()
        imgA=Input(shape=shape)
        imgB=Input(shape=shape)
        featA=extractor(imgA)
        featB=extractor(imgB)
        defeuclidean_distance(vectors):
        (featA,featB)=vectors
        sum_squared=k.sum(k.square(featA-featB),axis=1,keepdims=True)
        returnk.sqrt(k.maximum(sum_squared,k.epsilon()))
        distance=Lambda(euclidean_distance)([featA,featB])
        print(distance)
        outputs=Dense(1,activation="sigmoid",name='ddd')(distance)
        model=Model(inputs=[imgA,imgB],outputs=outputs)
        model.summary()
```

on generating pairs of images of same and different persons labeling same face image as 1 and different face as 0. We can train the euclidean layer to maximize the distance when the pair of faces is of same person otherwise minimize the distance.

## 5. RESULTS

The evaluation metric used in this experiment is accuracy. While training, we can see that oou model performed very accurate. By changing activation funtion and making the layers trainable our VGGFace transfer learning model achieved 99.42% accuracy on validation set. The results are given below:

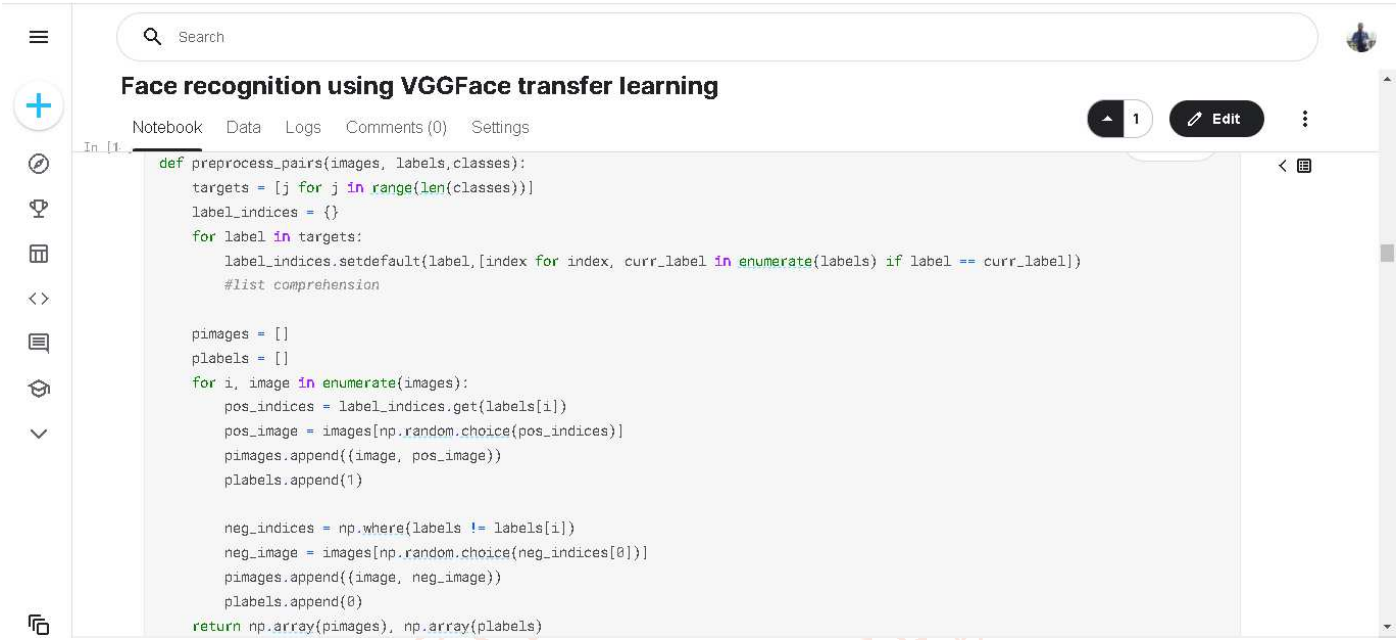


**Figure 2 Generating image pairs**

After no. of epochs the loss, accuracyis given as...

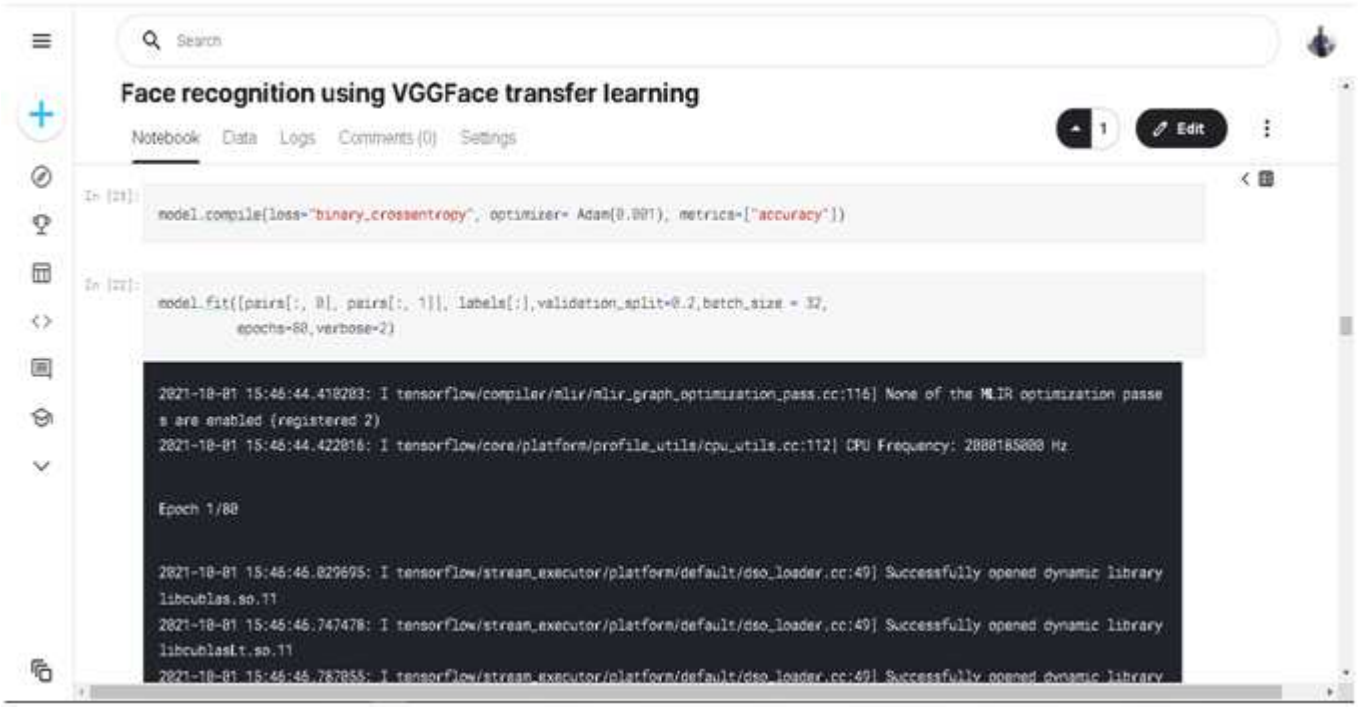


**Figure 3 Model fit function and few recorded epochs**

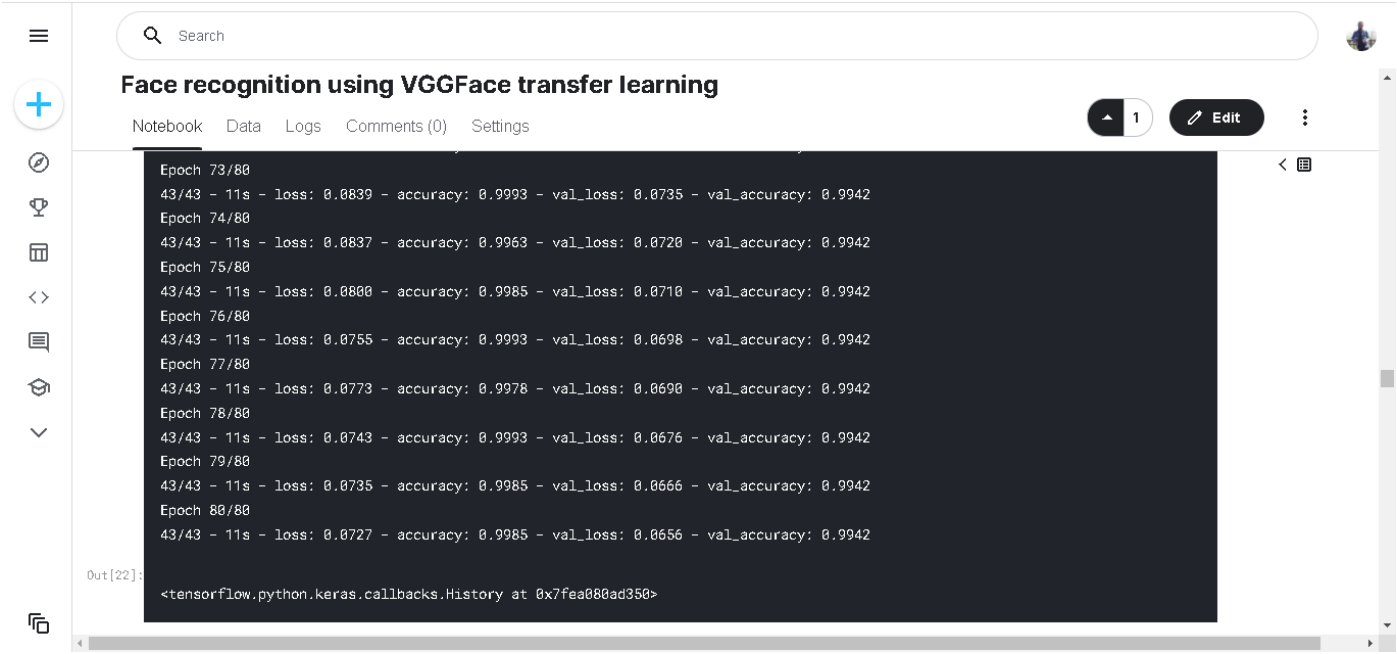


**Figure 4loss, accuracy after 80 epochs**

The corresponding curves of losses and accuracies were plotted:

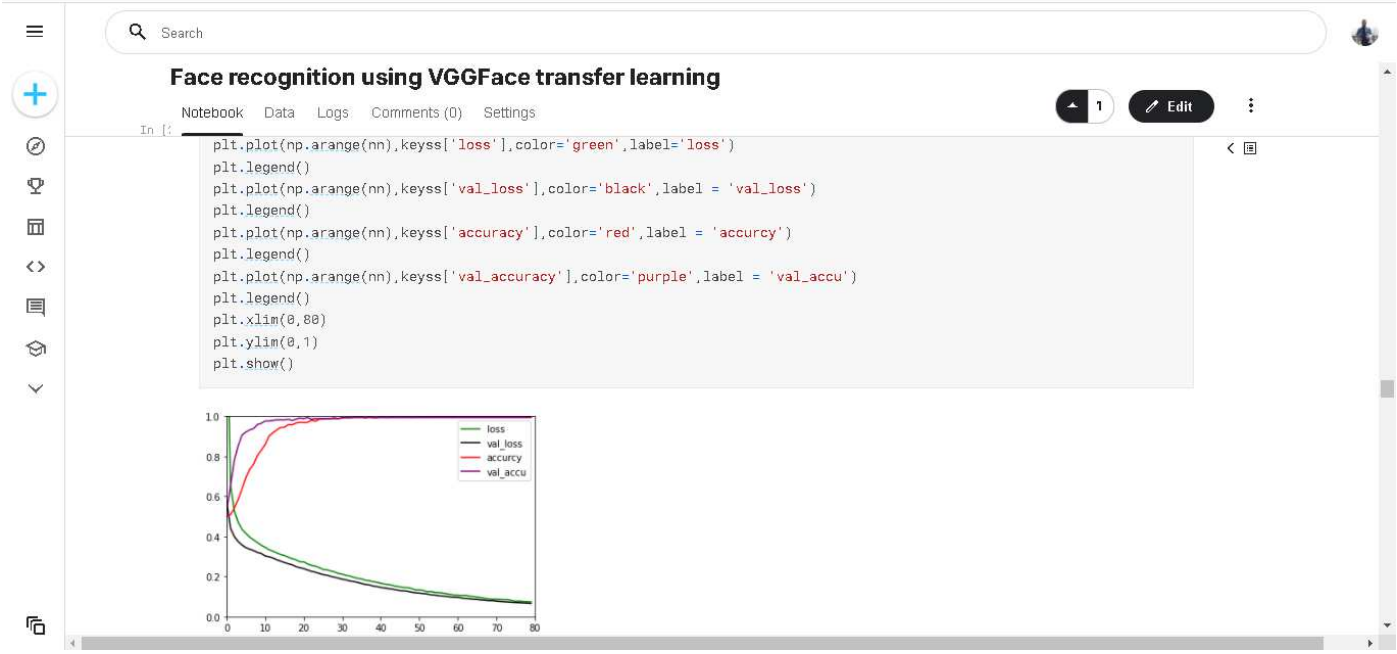


**Figure 5 loss, accuracy vs epochs plot**

The related notebook can be found here Click here.

## 6. CONCLUSION

In this paper, we showed experimentally and got an indication that using transfer learning on VGGFace we can improve the accuracy to recognize faces. Here the main research was about changing the activation function of from 34$^{th}$ layer of VGGface model to tanh. This is very promising as it is very tedious to collect a huge amount of data for training the model that has a high accuracy from scratch. For future works, We encourage vision researchers to explore more towards such techniques and add on how to make such methods more efficient.

## REFERENCES

[1] Wang Mei and Weihong Deng. Deep face recognition: A survey. arXiv preprint arXiv: 1804.06655, 2018.

[2] Mahbub Hussain, Jordan Bird, and Diego Faria. A study on cnn transfer learning for image classification. 06 2018.

[3] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Conference on fairness, accountability and transparency, pages 77–91, 2018.

[4] Sergio Escalera, Mercedes Torres Torres, Brais Martinez, Xavier Baró, Hugo Jair Escalante, Isabelle Guyon, Georgios Tzimiropoulos, Ciprian Corneou, Marc Oliu, Mohammad Ali Bagheri, et al. Chalearn looking at people and faces of the world: Face analysis workshop and challenge 2016. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 1–8, 2016.

[5] Kaipeng Zhang, Lianzhi Tan, Zhifeng Li, and Yu Qiao. Gender and smile classification using deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 34–38, 2016.

[6] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015.

[7] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), pages 17–24. IEEE, 2017.

[8] R. M. Prakash, N. Thenmoezhi, and M. Gayathri. Face recognition with convolutional neural network and transfer learning. In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), pages 861–864, 2019.