Face Recognition Using Deep Learning

J.Ashok Kumar[1], K. Sai Niavs[2], K. Srimanth[3]K. Sai Santosh[4], K. Nikhil Reddy[5]
[1, 2, 3, 4, 5] B V Raju Institute of Technology

Abstract- *Face detection and recognition in unconstrained environment are challenging due to various poses, illuminations and occlusions.However, conventional methods could no longer satisfy the demand at present, due to its low recognition accuracy and restrictions of many occasions. In this paper, we presented the deep learning method to achieve facial detection and face recognition. To solve the face recognition problem, we propose a deep cascaded Multi-Task Framework which exploits the inherent correlation between detection and alignment to boost up their performance. In particular, our Framework leverages a cascaded architecture with three stages of carefully designed deep convolutional networks to predict face in a coarse-to-fine manner.we present a system, called FaceNet, that directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced,tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors.*

Our method uses a deep convolutional network trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer as in previous deep learning approaches. To train, we use triplets of roughly aligned matching / non-matching face patches generated using a novel online triplet loss method and soft max is also used for face verification. The benefit of our approach is much greater representational efficiency: we achieve state of the art face recognition performance using only 128-bytes preface. With the implementation of both Multi-task Framework, FaceNet and Tensor Flow we are creating a model for a better Face Recognition.

Keywords- CNN, Face detection, Face recognition, Tensor flow

I. INTRODUCTION

Face recognition is one of the many wonders that AI research has brought forward to the world. It is a subject of curiosity for many researchers. Face recognition has received substantial attention from researchers due to human activities found in various applications of security like airport, criminal detection, face tracking, forensic etc. Compared to other biometric traits like palm print, Iris, finger print etc., face bio metrics can be non-intrusive. They can be taken even without user's knowledge and further can be used for security based applications like criminal detection, face tracking, airport security, and forensic surveillance systems.

Face recognition involves capturing face image from a video or from a surveillance camera. They are compared with the stored database. Face biometrics involves training known images, classify them with known classes and then they are stored in the database. When a test image is given to the system it is classified and compared with stored database. Face biometrics is a challenging field of research with various limitations imposed for a machine face recognition like variations in head pose, change in illumination, facial expression, aging, occlusion due to accessories etc.,. Various approaches were suggested by researchers in overcoming the limitations stated.Generally, how does machine can recognize faces?

To understand how a machine can recognize faces, we can start with asking ourselves how do we recognize a face? Most images of human faces have two eyes, a nose, lips, forehead, chin, ears, hair… That rarely changes. Yet, faces are different from each other. What makes them different? At the same time, face of the same person changes with emotion, expression, age… In fact just change in orientation creates a different image. How do we identify a person in spite of all that?

On a gross level, one can say that there are some components of a face are related to age, emotion and orientation. While there are some other components that are stick to the person irrespective of the age, emotion, etc. But, one can say that there are several overlapping components of the face which are individually responsible for the perception of emotion, age and the person himself. Essentially, we know that there is "some relation that is too complex for logic" that is where deep learning shows up!

Fig 1: Face detection and face extraction of a image.

In this paper, we propose the face recognition system using tensor flow framework, facenet, MTCNN(Multi Task Convolutional Neural Network), CNN(Convolutional Neural Network).In this project, we are using deep learning algorithms instead of machine learning algorithms.Because machine learning algorithms consume more memory than deep learning algorithms.Moreover, accuracy level of machine learning algorithm is low.So, to improve accuracy level of this project we are using deep learning.

II. RELATED WORK

As we know there are many research work is going on and done on face recognition. And there are many algorithms for face recognition in deep learning and machine learning.

Face recognition algorithms are broadly classified into two classes as image template based and geometric feature based. The template based methods compute correlation between face and one or more model templates to find the face identity. Principal component analysis, linear discriminate analysis, kernel methods etc. are used to construct face templates. The geometric feature based methods are used to analyze explicit local features and their geometric relations (elastic bung graph method). Multi resolution tools such as contour lets, ridge lets were found to be useful for analyzing information content of images and found its application in image processing, pattern recognition, and computer vision.

Numerous algorithms have been proposed for face recognition; Chellappa et al (1995), Zhang et- al (1997) and Chan et al (1998) use face recognition techniques to browse video database to find out shots of particular people. Haibo Li et al (1993) code the face images with a compact parameterized facial model for low-bandwidth communication applications such as videophone and teleconferencing. Recently, as the technology has matured, commercial products have appeared on the market.

Turk et al (1991) developed Principal Component Analysis (PCA) technique for Face recognition to solve a set of faces using Eigen values. Rama Chellappa et al (2003) have dealt with the feature based method using statistical, structural and neural classifiers for Human and Machine Recognition of Faces. Krishnaswamy et al (1998) proposed automatic face recognition using Linear Discriminant Analysis (LDA) of Human faces.

Chengjun Liu and Harry Wechsler (2002) presented new coding schemes, the Probabilistic Reasoning Modes (PRM) and Enhanced Fisher linear discriminant Models (EFM) for indexing and retrieval from large image databases. Michael Bromby (2003) has presented a new form of Forensic identification-facial biometrics, using computerized identification.

Joss Beveridge et al (2003) provided the PCA and LDA algorithms for face recognition. A detailed Literature Survey of Face Recognition and Reconstruction Techniques were given by Roger Zhang and Henry Chang (2005).

Vytautas Perlibakas (2004) has reported a method in “Face Recognition Using Principal Component Analysis and Wavelet Packet Decomposition” which allows using PCA based face recognition with a large number of training images and performing training much faster than using the traditional PCA based method.

III. APPROACH

As mentioned above we are using tensorflow, facenet, MTCNN,CNN for face recognition. In this section you will know how are we going to using these mentioned algorithms and framework in this project.

In this project we are using many libraries which are numpy, opencv, scikit-learn, spciy. We using CNN a deep learning algorithm, MTCNN, FaceNet, Tensorflow framework.

A. MTCNN:

In this project, we are using MTCNN for face detection and face recognition.So, MTCNN is Multi Task Convolutional Neural Network, which is an algorithm

consisting of 3 stages, which detects the bounding boxes of faces in an image along with their 5 Point Face Landmarks.Generally MTCNN consists of three Steps which are

1. Identify, crop and align face
2. Generate embeddings
3. Compare embeddings

Stage 1 : Obviously, the first thing to do would be to pass in an image to the program. In this model, we want to create an image pyramid, in order to detect faces of all different sizes. In other words, we want to create different copies of the same image in different sizes to search for different sized faces within the image.

In the P-Net, for each scaled image, a 12x12 kernel runs through the image, searching for a face. In the image below, the red square represents the kernel, which slowly moves across and down the image, searching for a face.Within each of these 12x12 kernels, 3 convolutions are run through with 3x3 kernels. After every convolution layer, a prelu layer is implemented (when you multiply every negative pixel with a certain number 'alpha'. 'Alpha' is to be determined through training). In addition, a maxpool layer is put in after the first prelu layer(maxpool takes out every other pixel, leaving only the largest one in the vicinity).

After the third convolution layer, the network splits into two layers. The activation from the third layer are passed to two separate convolution layers, and a softmax layer after one of those convolution layers (softmax assigns decimal probabilities to every result, and the probabilities add up to 1. In this case, it outputs 2 probabilities: the probability that there is a face in the area and the probability that there isn't a face).

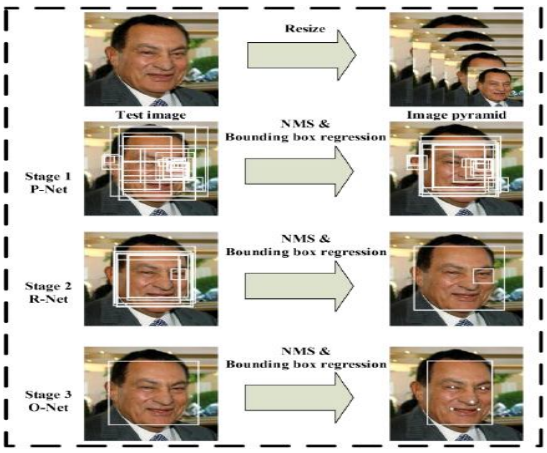

Fig 2: Pipeline of our cascaded framework that includes three-stage multi-task deep convolutional network

Stage 2: Sometimes, an image may contain only a part of a face peeking in from the side of the frame. In that case, the network may return a bounding box that is partly out of the frame.After we pad the bounding box arrays, we resize them to 24 x 24 pixels, and normalize them to values between -1 and 1. Currently, the pixel values are between 0 to 255 (RGB values). By subtracting each pixel value by half of 255 (127.5) and dividing it by 127.5, we can keep their values between -1 and 1.R-Net's output is similar to that of P-Net: It includes the coordinates of the new, more accurate bounding boxes, as well as the confidence level of each of these bounding boxes.Once again, we get rid of the boxes with lower confidence, and perform NMS on every box to further eliminate redundant boxes. Since the coordinates of these new bounding boxes are based on the P-Net bounding boxes, we need to convert them to the standard coordinates.After standardizing the coordinates, we reshape the bounding boxes to a square to be passed on to O-Net.

Stage 3: Before we can pass in the bounding boxes from R-Net, we have to first pad any boxes that are out-of-bounds. Then, after we resize the boxes to 48 x 48 pixels, we can pass in the bounding boxes into O-Net.The outputs of O-Net are slightly different from that of P-Net and R-Net. O-Net provides 3 outputs: the coordinates of the bounding box (out[0]), the coordinates of the 5 facial landmarks (out[1]), and the confidence level of each box (out[2]).Once again, we get rid of the boxes with lower confidence levels, and standardize both the bounding box coordinates and the facial landmark coordinates. Finally, we run them through the last NMS. At this point, there should only be one bounding box for every face in the image.

B. FaceNet:

FaceNet is a Deep Learning architecture consisting of convolutional layers based on GoogleNet inspired inception models. FaceNet returns a 128 dimensional vector embedding for each face.FaceNet takes an image of the person's face as input and outputs a vector of 128 numbers which represent the most important features of a face. In machine learning, this vector is called embedding.

In this project, we use facenet for training the image. Facenet follows following steps to train the images.

i. Randomly selects an anchor image(any image in dataset).
ii. Randomly selects an image of the same person as the anchor image.
iii. Randomly selects an image of a person different than the anchor image.

iv. Adjusts the FaceNet network parameters so that the positive example(same person) is closer to the anchor than the negative example(different person).

So, in the training process, facenet keeps all images of one person at same place from different images in dataset.Anchor image is kept closer to positive case than negative case as shown in below figure.

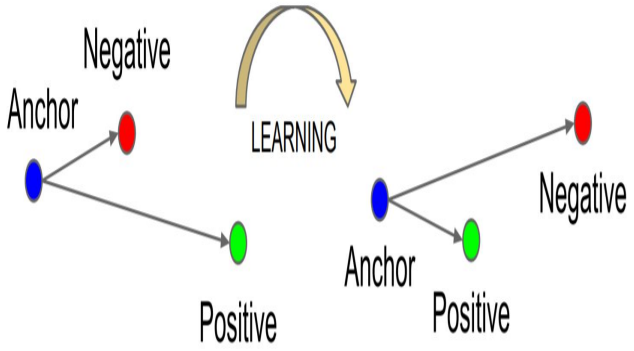

Fig 3: Training Process of images

Triplet Loss:

Another important aspect of FaceNet is its loss function . It uses triplet loss function. In order to calculate the triplet loss, we need 3 images namely anchor, positive and negative.

The intuition behind triplet loss function is that we want our anchor image (image of a specific person A) to be closer to positive images (all the images of person A) as compared to negative images (all the other images).

Triplet loss function can be formally defined as

$$\sum_{i}^{N} \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

Fig 4: Formula of triplet loss

Soft Max:

We mentioned earlier that the classification step could be done by calculating the embedding distances between a new face and known faces, but that approach is too computationally and memory expensive. Instead, we decided to use the Softmax classifier which memorises boundaries between people which is much more efficient. Softmax classifier is used as a final step to classify a person based on a face embedding. Softmax was a logical choice for us since the entire stack is neural networks based. If the face embeddings themselves are good, all classifiers should perform well at this step.

C. CNN:

In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

In this project, facenet trains CNN.FaceNet uses deep convolutional neural network (CNN). The network is trained such that the squared L2 distance between the embeddings correspond to face similarity.FaceNet trains CNN using Stochastic Gradient Descent (SGD) with standard backprop and AdaGrad.

Stochastic Gradient Descent (SGD) is an optimization technique that is used to optimise our loss function.The two axes (x and y) represent weights and the third axis (z) represents the loss with respect to those two weights.

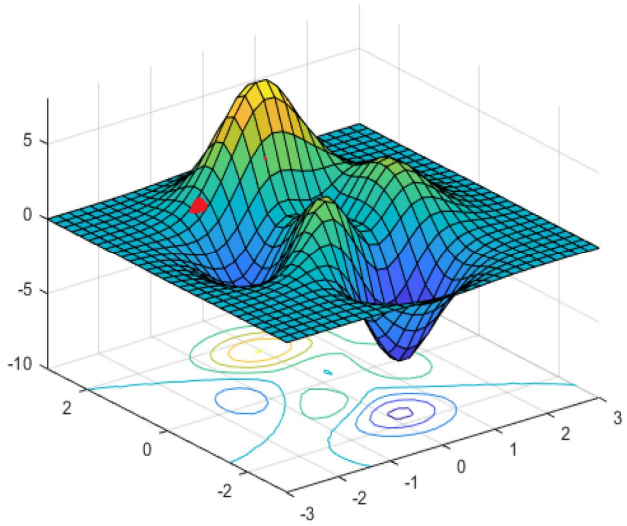

Fig 5: SGD

AdaGrad is used to generate variable learning rates. Fixed learning rates do not work well in deep learning. In case of CNN where each layer is used to detect a different feature (edges, patterns etc.), a fixed learning will just not work, as different layers in our network require different learning rates to work optimally.

D. TensorFlow:

TensorFlow is an open source library for numerical computation and large-scale machine learning. TensorFlow bundles together a slew of machine learning and deep learning models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework. TensorFlow was created by "GOOGLE BRAIN TEAM".

In this project tensorflow is used by MTCNN and CNN. It is used in training of algorithm to identify the image. We use TensorFlow's Queue API to load the preprocessed images in parallel using multi-threading.

CNN can be quite slow due to the amount of computations required for each iteration. we'll now use GPU's to speed up the computation.Tensorflow, by default, gives higher priority to GPU's when placing operations if both CPU and GPU are available for the given operation. When using a GPU, this allows image preprocessing to be performed on CPU, while matrix multiplication is performed on GPU.

IV. RESULT

We have done face recognition using machine learning we have output of that project. In this project we have output of this project. We have compare the accuracy of these project.

And we have made graph to represent it which is shown below.

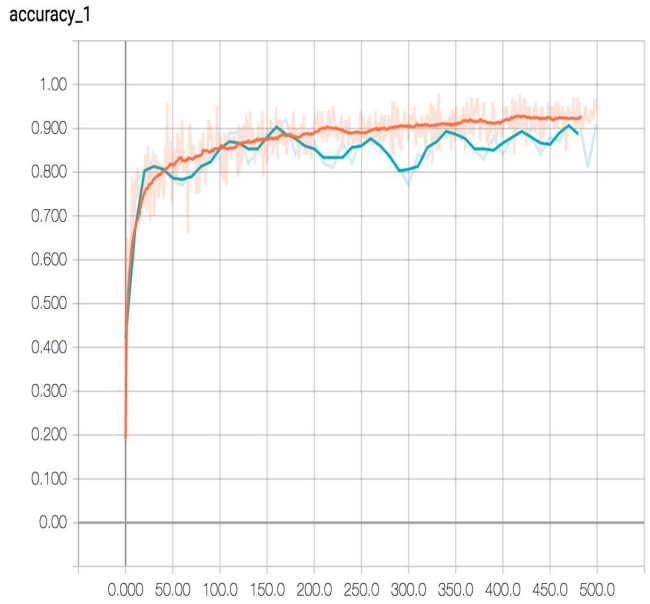

Fig 6: Graph between accuracy of face recognition using deep learning and face recognition using machine learning

In this graph,the orange line shows the accuracy of the model on the face recognition using deep learning . While the blue line shows the accuracy on the face recognition using machine learning .

Also by using proposed system we have recognize the image identity. In this we have recognize the face in both image and the video which are shown below.

In this section , We conduct experiments to verify the effectiveness of our proposed Face Recognition model All the experiments were performed by NVIDIA GEFORCE GPU using Multi-Task Convolutional neural network.We are considering detection rate of MTCNN using the parameter threshold as threshold = [0.6,0.7,0.7] and scaling factor is set to 0.709 . Apart from threshold and frame_interval,image_size,input_image_size,margin,mini-size are taken for accuracy of image detection and saving of cropped images. Experiments are conducted to recongize face accurately without any flaws.

In the Proposed model three Datasets are taken for creating face recognition model. Faces inside the image must be completely visible .So that there are no errors in detection . All the images taken are in jpg format . A sample image is shown below . Finally , with the help of facenet model we are able to train new model based on our datasets which result to classifier.pkl file . Which is given as input to video which have taken or image .

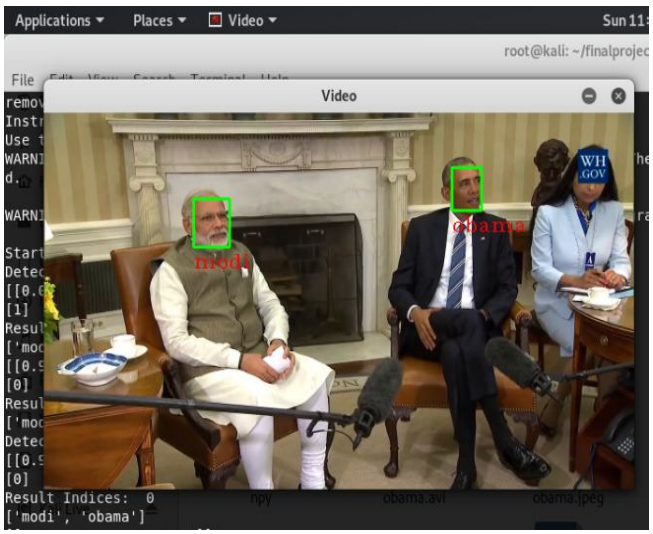

Fig 7: Recognition of face through video using proposed system.

V. CONCLUSION

We have developed a fully working Face Recognition system, with the capacity of online learning. It can work with

any kind of images and videos, and is reasonably robust to changes in face expression or orientation, light conditions and other factors.

The other application consists in recognizing people in videos. After processing the video, it draws the bounding box of each person in it, following them around the screen, and writes the name of identified people. It is already working for videos with multiple people in it.

The obtained system has been extensively tested, and different parameters combinations have been tried. We have taken a random video from google ,of Modi & Obama and sent the dataset to assess the performance of our face verification step.

Apart from that, we have gathered a 300 face images dataset to use both in training and testing. The facial recognition part has been tested using two different sized datasets, and we have obtained steady results around the 90% of accuracy, reaching a maximum of 95%. These results are better than the ones we expected, and they allow for some real life use cases. However, there is still room for improvement.

We have done face recognition using tensorflow framework, facenet, MTCNN, CNN and identify the face through image and videos with high accuracy.

REFERENCES

[1] Florian Schroff, Dmitry Kalenichenko, James Philbin,FaceNet: A Unified Embedding for Face Recognition and Clustering, Google Inc, June 2015.
[2] Arun Mandal, MTCNN Face detection and face matching using tensorflow and facenet, February 2018.
[3] Luka Dulčić, Face recognition using facenet and MTCNN, arsfutura magazine, Nov 2019.
[4] Dhariya Kumar, Introduction to FaceNet: A Unified Embedding for Face Recognition and Clustering, July 2019.
[5] Sumit Saha, A Comprehensive guide to convolutional neural networks-the Eli5 way, June 2018.
[6] Chi-Feng Wang, How does face detection program works using neural networks, July 2018.