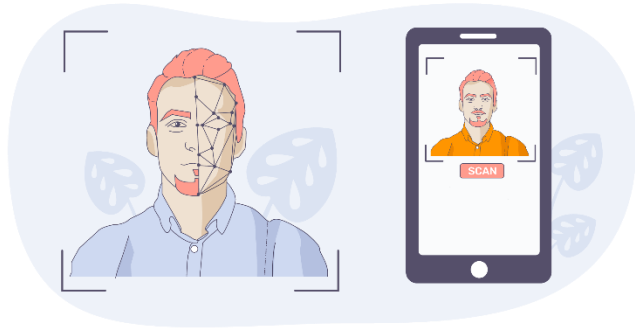# Deep Learning Face Recognition Report (WiDS Project)

Desai Utsav Manojkumar

Roll no.: 200100054

Mechanical Engineering

Introduction

Face Recognition is one of the hot topics in the field of Deep Learning. In this project, we aim to develop a Deep Learning model using the TensorFlow library of python.

I started by reading previous studies and research papers on Face Recognition in Deep Learning. These papers include FaceNet, DeepFace, and ArcFace. As a part of learning, I also completed various online courses from Coursera, an online learning platform, and some courses from IITB. Coursera courses include 'Neural Networks and Deep Learning' and 'Convolutional Neural Networks,' both delivered by Andrew Ng, co-founder of Coursera. Also, I did a course on 'Image Processing' by Professor Amit Sethi from IIT Bombay in my 5th semester.

Along with these courses, I also completed some projects related to Deep Learning, which include guided projects like 'Image Segmentation using PyTorch' and 'Facial Expression Recognition' and a self-project on 'Hand Gesture Recognition.'

Model Description

In this model, I used a ResNet50 base model stack with fully connected layers on it. The image size was 112×112×3, where 3 represents RGB channels. The dataset I used for this project is the Labeled Faces in the Wild Home (LFW) dataset from Kaggle. It contained a total of 13,233 images of 5,749 people, of which 1680 people pictured have two or more distinct photos in the dataset.

Further, talking about the architecture of the model; first and foremost is a ResNet50 model with pre-trained weights from 'ImageNet.' Also, the weights of the ResNet50 part were never modified in training. The ResNet50 part is followed by a GlobalAveragePooling2D layer. Then we flatten the output of this pooling layer using the Flatten layer. These layers are followed by two Dense layers, both of size 512, which are finally followed by the output layer, lambda, also of size 512.

The optimized values of hyperparameters found using the hit and trial method are: number of epochs = 10, learning rate = 0.001, and no batches are used for training, i.e., the dataset as a whole was considered at a time. The optimizer used was Adam, and the loss function used here was TripletSemiHardLoss. Also, I used haarcascade to improve the dataset; basically, what haarcascade does is extract only the face part from the image, which is highly desirable for improving the accuracy. The final loss obtained on the whole dataset is 0.7943. The below graph shows the loss variation with the epochs.
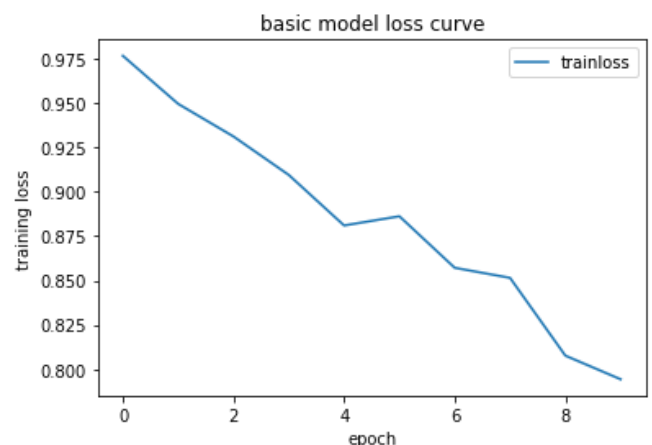
References and Links

[FaceNet Paper](), [DeepFace Paper](), [ArcFace Paper]()

GitHub Repository: [Deep_Face_WiDS]()

[Image_1](), Image_2(Graph: from the Google Colab notebook)