CS GY 9223: Deep Learning

Face Recognition and Emotion recognition for Sentiment Analysis

Aishwarya Kore (adk497)   Tejas Shetty (trs389)

Fall 2020

Abstract

Recent advances in face recognition models have led to powerful implementations that provide highly accurate results over datasets that have a high volume of images. In this paper,we have built a model by cascading two popular models, Facenet [1] and VGGface [2] to perform face recognition and emotion detection respectively. We used a pretrained model for face recognition and ran it on the 5 celebrity faces dataset [10] giving accuracy of 95%+. The emotion detection model was trained and tested on the FER2013 [11] to obtain an accuracy of 67% on the test set. We were motivated to design a unique model that is a combination of face recognition and emotion detection. While we have individual models, combining both of them gives a labelled representation of the emotions exhibited by a person. Such a model can be primarily used in the field of sentiment analysis with applications ranging from detecting the sentiment of multiple people at the same time to detecting the sentiment of the same person over time. For both these areas, outputs obtained from this model would serve as the basis to derive insights.

1 Introduction

In this paper we present a unified system for sentiment analysis by cascading the Facenet [1] model for face recognition and the VGGface [2] model for emotion detection. Our combined model basically takes an input image and gives two outputs: the first being the identity and the second his/her emotion. This model would produce real time results, for a given input image. Such a real-time facial recognition and emotion detection has potential uses in many applications and industries.For example, corporate can use the model to identify an employee, mark their presence and note their moods regularly. This can help the authorities to evaluate the collective mood of all the employees that day so that they can predict the productivity, and also to provide the needed support or resources for the employees who need it.Similarly, if this model is applied to multiple people at the same time, it gives an overall indication of the response of a cohort to an event. e.g the student's reaction to an announcement by the professor and so on.

2 Literature Survey

We divide this section into the related work done for the face recognition and emotion detection:

2.1 Face Recognition

There has been a lot of research on face recognition with a wide variety of models available. We here describe the two most popular models, Facenet [1] and Deepface [2] along with other models like AIM [4] (Age Invariant Model)

Facenet[1] by Florian Schroff et. al. is a Deep Neural network that learns mappings from a human face to create a single n dimensional vector representing the face. For example, given a 160*160 image, it produces a 128 d embedding. Facenet uses the triplet loss function to learn similar and dissimilar images from an embedding. It uses an anchor, a positive and a negative image and finds a threshold value so the positive and the negative images can be classified w.r.t. the anchor.The internal deep architecture used in Facenet is of two types: Zeiler & Fergus inspired architecture and the inception inspired architecture. Their practical differences lie in the difference of parameters and FLOPS. Once we have a trained model, we can use it for labelling by first creating the embeddings and then calculate the distance between them.The Facenet model has an accuracy of 99.63% on the LFW dataset and 95.12 % on the You Tube Faces dataset.

Deepface[3] by Yaniv Taigman et.al. is also a deep neural network model open sourced by Facebook. It consists of a 3D face modelling followed by piecewise affine transformatio. The final face representation is derived from a 9 layer Neural network. The deep face model is not affected by the alignment or lighting on the picture.The Deep Face neural network consists of a series of pooling-convolution layers followed by fully connected layers. The probability of each class is maximized by the cross entropy loss. Deep face has an accuracy of 97.25% on the LFW dataset and 91.4% on the You Tube faces data set.

AIM [4] by Jian Zhao et.al. presents a novel unified deep architecture jointly performing cross-age face synthesis and recognition in a mutual boosting way. AIM achieves continuous face rejuvenation aging thus avoiding the requirement of paired data and the true age of testing samples. AIM extends from an auto-encoder based GAN, and consists of a disentangled Representation Learning sub-Net (RLN) and a Face Synthesis sub-Net (FSN) that jointly learn discriminate and robust facial representations disentangled from age variance and perform attention-based face rejuvenation/aging end-to-end. This model has several datasets to demonstrate its accuracy. The paper mentions a self created CAFR benchmark consisting of 1,446,500 images with various age variations. The model has an accuracy of 84.81% on this dataset.

2.2 Emotion Detection

There are various models for Emotion Detection as well with a large chunk of them using deep CNN based architectures. In our implementation as well we would be implementing a CNN based model for emotion recognition Prior to CNN based models, some notable techniques used in emotion recognition are given below: 1) pixel-based recognition , 2) local binary pattern , 3) wavelet transform , 4) discrete cosine transform , 5) Gabor filter , 6) edge and skin detection , 7) facial contour , and 8) fuzzy logic model.Each of these studies has shown that the facial emotion recognition can accomplish an average level of success, but the performance is less than human judgement. It was after the advent of CNN based results that we started obtaining state of the art results in the filed of emotion detection. Below we describe two CNN based models for emotion detection:

The model described in this chapter[5] by Alex D. Torres et.al discusses the use case of using facial emotion recognition for sentimental analysis of medical patients. This chapter also focuses on implementing the system in real time by using cloud platforms that have General Purpose GPU which accelerates the training of the model. The model described here is CNN that consists of 7 layers: 3 convolutional layers, 2 max-pooling layers, and 2 fully connected layers.The accuracy obtained by this model is 65% on their private test set. This chapter also discusses in depth about GPU benchmarking and comparison.

The model described in the paper[6] by Christopher Pramerdorfer et. al. uses an ensemble of differently trained classifiers to make the final facial emotion prediction. The final prediction is made by combining the predictions of all classifiers within the ensemble through a learned weighted average. The classifiers used to make up the ensemble all use facial landmark detection, some kind of facial registration, and various forms of illumination pre-processing. In order to obtain different classifiers on the same dataset, the state-of-the-art models employ different combinations of histogram equalization and linear plane fitting on the various classifiers used to compose the ensemble. Also, these two models both use illumination normalization that normalizes every input image to have a mean of 0 and norm of 100. By forming an ensemble of modern deep CNNs, this model obtains a FER2013 test accuracy of 75.2%, outperforming previous works without requiring auxiliary training data or face registration.

3 Dataset Description

We used the 5 celebrities dataset with the face recognition model and the FER2013 dataset to train the emotion detection model. Below are their respective details:

3.1 5-celebrities dataset

This dataset [10] consists of 14-20 photos each of the celebrities Ben Affleck, Elton John , Jerry Seinfeld , Madonna , Mindy Kaling. The validation directory has 5 photos of each celebrity.The photos haven't been cropped for consistent aspect ratios.Later on we also added our own data by adding in another class with our own pictures in training as well as validation data.

3.2 FER2013

As described in [11], the data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The "emotion" column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The "pixels" column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the "pixels" column. It consists of 28,709 training samples,3589 validation images and 3589 test images.

4 Model Description

Below is a flow diagram of our implementation with description of the neural network models used, and the operations performed on images.

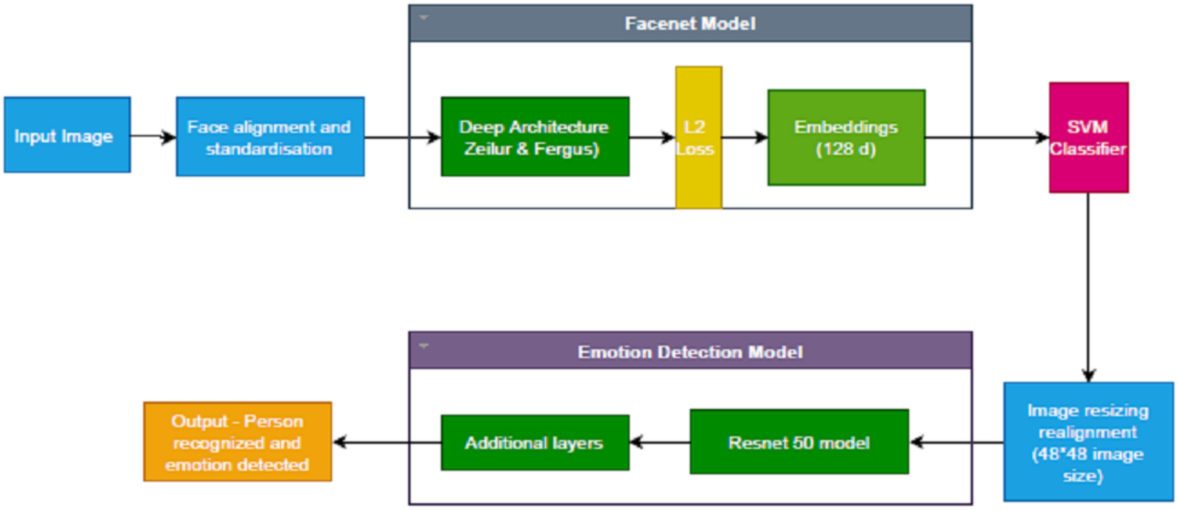

Facenet Model
Input Image
Face alignment and standardisation
Deep Architecture Zeilur & Fergus)
L2 Loss
Embeddings (128 d)
SVM Classifier
Emotion Detection Model
Output - Person recognized and emotion detected
Additional layers
Resnet 50 model
Image resizing realignment (48*48 image size)

Figure 1: System Flow - Block diagram

A detailed description of the components of the above diagram is presented below.

4.1 Facenet-Keras

The Facenet [8] paper published by Google mentions the Zeiler & Fergus architecture and the Inception based one In the paper, the Inception based architecture is based on Google based GooglLeNet based Inception models. These models though have similar performance to the Zeiler & Fergus model but have dramatically less computational needs. The Inception model has around 7.5M parameters with 1.6B of FLOPS (a metric for no of computations needed) compared to the Zeiler& Fergus based model that has 140M parameters and 1.6B FLOPS. In other variations, the FLOPS of the inception model come down to around 550M. The Keras implementation of Facenet which is used in this paper uses the inception-resnet-V1 based architecture. It has 447 layers. The json representation of the model[12] is included for further understanding. The Facenet network is trained via a triplet loss function that encourages vectors for the same identity to become more similar (smaller distance), whereas vectors for different identities are expected to become less similar (larger distance). The focus on training a model to create embeddings directly (rather than extracting them from an intermediate layer of a model) was an important innovation in this work. These face embeddings are then used as the basis for training the classifier systems on our dataset. The Facenet model can be used as part of the classifier itself, or we can use the Facenet model to pre-process a face to create a face embedding that can be stored and used as input to our classifier model. We are using this latter approach as it is preferred as the Facenet model is both large and slow to create a face embedding.We then pass those face embeddings to out SVM classifier model.

4.2 Linear Support Vector Machine (SVM)

It is very common to use SVM while working with normalized face embedding inputs. This is because the method is very effective at separating the face embedding vectors. We can fit a linear SVM to the training data using the SVC class in scikit-learn and setting the ‘kernel‘ attribute to ‘linear‘. SVM predicts the right class to which the input image belongs to.This gives us the person in the input image.

4.3 VGGFace Model Resnet50

Keras has a VGGFace library that provides us three networks to use the VGGFace (based on VGG16 architecture), Resnet(Based on Resnet 50 architecture) and Setnet(based on Setnet 50 architecture. We have used the Resnet 50 model in our use case. The Resnet architecture consists of a skip connection where the weights from a layer are connected to the next layer as well as next to next layer. This helps primarily to solve the problem of vanishing gradients during back propagation. It consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters. We are using the Resnet-50 model for emotion detection and add to it three layers and ReLu activation function in and finally softmax activation function layer as a classifier at the end. We have used Adam optimizer and categorical cross entropy loss while training the model.The input to this model is through a custom data loader with a batch size of 128.

5 Training Details and System Implementation

We know that Facenet is pretranied on tens of millions of images as mentioned in [8]. We get the face embeddings of the data images directly by using getEmbedding() method defined in Facenet model.We store those embeddings in .npz file that we can use later. We encode the labels, normalise and transform the training data embeddings then fit them on the linear SVC. This way our classifier learn the different classes and predicts the right class for the test dataset. For emotion detection we train a sequential Resnet-50 model on 28709 images and we validate it on 3589 images. We further use 3589 images to test the model. The training is done over varying epochs(5,10,20) and learning rates(default,1e-04,1e-05) to find the best combination which is further discussed in the Section 6. Let us walk through the system designed by us step by step:

1. We input an image to the system. Its format can be .png or .jpeg

2. The face recognition performs face detection by cropping the image around the face and extracting only the face to be recognized from the entire image using bounding boxes. We resize the pixels of the extracted image to the model size which is 160*160 in our case. This process is labelled as face alignment and standardisation.

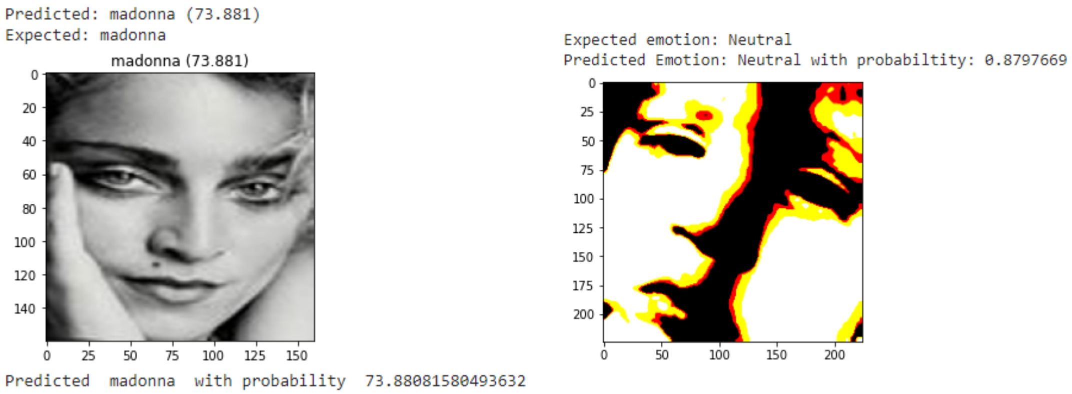
3. We use a pre-built keras model for face recognition built on the Facenet framework using the keras api available as an h5 file. We perform transfer learning on the this model on using the dataset that has images belonging to 5 different classes/people.It converts each face into an embedding which means that we extract high-quality features from each face and produce a 128 dimensional vector representation of the same. This vector is normalized and the vector size can be changed from 128 to 1024. Faces with similar face embeddings have vectors that are closer to each other.

4. We are using Linear SVM as the classifier model. The face embeddings that are similar to each other are clustered together and the model predicts the right class that the face belongs to

5. For emotion detection the model requires images to be of 48x48 pixel and a grayscale image. So we did some image pre-processing like image resizing, realignment etc. again on our input image so that it can be used in our emotion detection model.

6. We then predict the emotions that are depicted in the image.

7. The repository link to our model implementation can be found in the references [13]

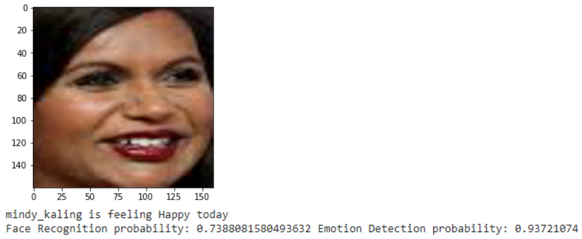


```
Predicted: madonna (73.881)
Expected: madonna
```

```
Predicted  madonna  with probability  73.88081580493632
```

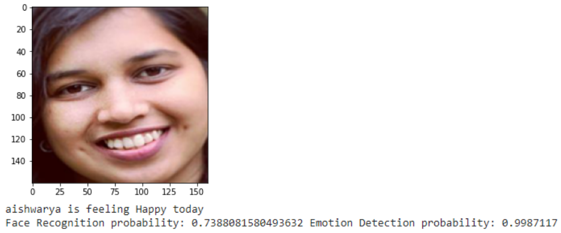
```
Expected emotion: Neutral
Predicted Emotion: Neutral with probabiltity: 0.8797669
```

Figure 2: Results of the Face Recognition and Emotion Detection models respectively



```
mindy_kaling is feeling Happy today
Face Recognition probability: 0.7388081580493632 Emotion Detection probability: 0.93721074
```

Figure 3: Output of our system for arbitrary images of a celebrity from [10] dataset



```
aishwarya is feeling Happy today
Face Recognition probability: 0.7388081580493632 Emotion Detection probability: 0.9987117
```

Figure 4: Output of our system for arbitrary images apart from the [10] dataset.

6 Results And Observations

The face recognition model predicts the right class that the face belongs to with an accuracy of 95+% every time. The emotion detection model gives accuracy of 67%. We have used categorical cross-entropy loss while training this model. By varying the learning rate hyper parameter we notes that the learning rate that gives the best accuracies and least losses in both training and testing, using Adam optimizer is 1e-4. Observations for different learning rates are shown in the figures below and Table 1. According to Table 2. we also notices that beyond ten epochs the accuracy does not vary a lot and the loss increases so we did early stopping after 10 epochs To get more insights about our results we plotted a confusion matrix between the predicted and true labels for our test dataset which is shown in Figure 8. We found that we could predict "Happy" emotion correctly 87 % of the times which was the highest correctly classified emotion.We mis predicted "Fear" emotion the most. Fear was mostly classified as Sad and disgust was mis classified as sad and angry which can be understandable that these emotions could get interchanged. Overall we got very good accuracies for our system and got insights on the where there could be scope of improvement.

Learning Rate	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
Default	0.8569	0.6560	0.4090	1.1817
1e-04	0.8193	0.6585	0.5080	0.9861
1e-05	0.6140	0.5739	1.0534	1.1543

Table 1: Accuracy and Losses for different Learning Rates

No.of Epochs	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
5	0.8193	0.6585	0.5080	0.9861
10	0.9155	0.6727	0.2485	1.1310
20	0.9618	0.6763	0.1133	1.3898

Table 2: Accuracy and Losses for lr= 1e-04

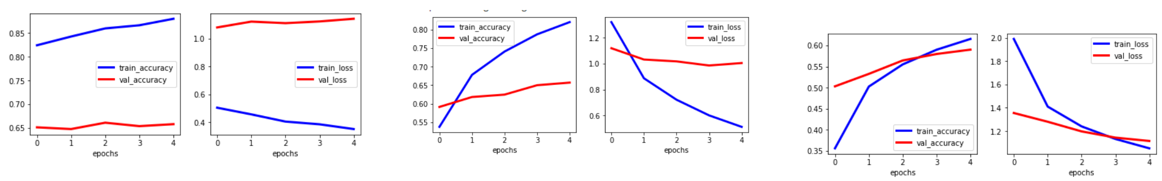


Figure 5: Training Accuracy,Validation Accuracy vs epochs and Training Loss,Validation Loss vs epochs for default lr, lr=1e-4, lr=1e-5 respectively

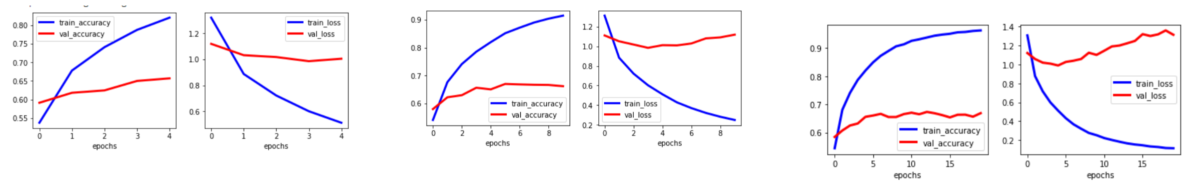


Figure 6: Training Accuracy,Validation Accuracy vs epochs and Training Loss,Validation Loss vs epochs for default lr=1e-4 for epochs=5,epochs=10,epochs=20 respectively

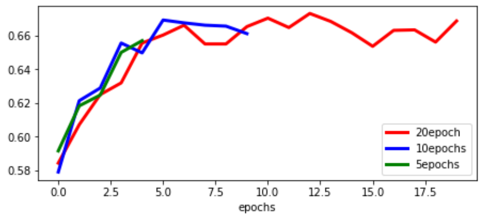
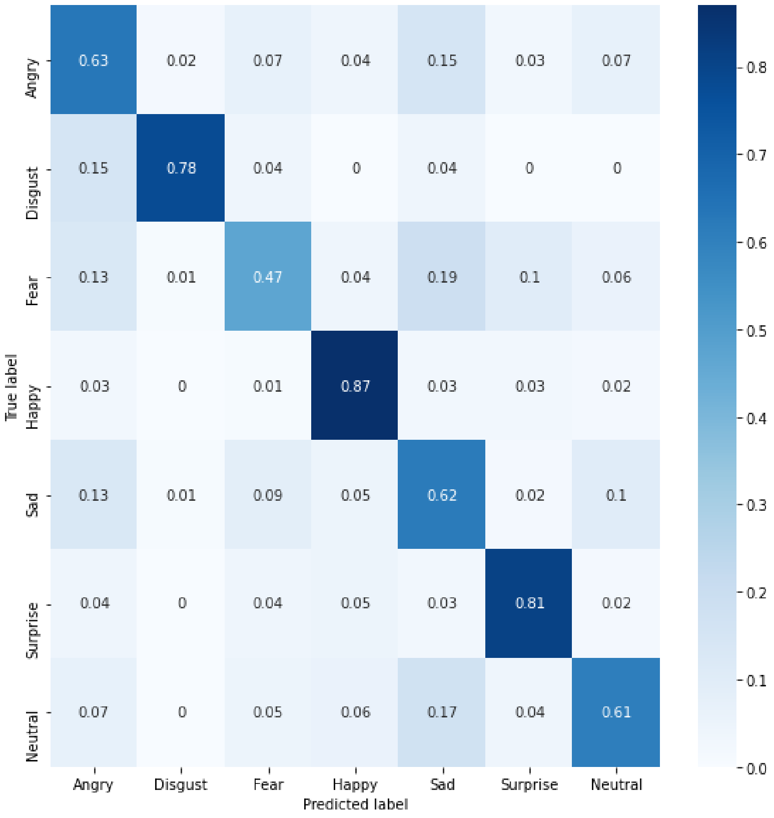


Figure 7: Validation Accuracy vs epochs for lr=1e-4 for epochs=5,epochs=10,epochs=20



	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	0.63	0.02	0.07	0.04	0.15	0.03	0.07
Disgust	0.15	0.78	0.04	0	0.04	0	0
Fear	0.13	0.01	0.47	0.04	0.19	0.1	0.06
Happy	0.03	0	0.01	0.87	0.03	0.03	0.02
Sad	0.13	0.01	0.09	0.05	0.62	0.02	0.1
Surprise	0.04	0	0.04	0.05	0.03	0.81	0.02
Neutral	0.07	0	0.05	0.06	0.17	0.04	0.61

Figure 8: Confusion matrix for Emotion Detection Results

7 Conclusion and Future Work

We provide a well organised code that can detect and recognise a particular person and predict the emotions of this person at that very instance. We also tested this system on people other than the ones given in [10]. We just needed to add 5-10 images of a new person in a new folder in the dataset and the system accurately identified this new person and emotions of that person. We came across interesting findings about the Hyper Parameter as mentioned in section 6 and would recommend using these to save time and resources while using the VGGFace resnet 50 model.We found out that certain emotions like fear,disgust was confused with sadness and anger respectively so we suggest including more images of these emotions while training the model. This code can be used for further development of a live system that can stream live video and detect the people and their moods.Various live streaming technologies can be used like AWS Kinesis Video Stream, Kinesis Data Streams, Rekognition in combination with Sage Maker Studio to deploy this model. We successfully combined two different models to create a well functioning system and achieved our goal of creating a working code that is highly accurate for Face Recognition and Emotion recognition for Sentiment Analysis.

References

[1] Jason Brownlee. How to develop a face recognition system using facenet in keras.Retrieved from https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/

[2] VGGFace model for Keras https://gist.github.com/EncodeTS/6bbe8cb8bebad7a672f0d872561782d9

[3] Yaniv Taigman,Ming Yang,Marc'Aurelio Ranzato,Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. Retrieved from https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf

[4] Jian Zhao, Yu Cheng, Yi Cheng, Yang Yang, Haochong Lan, Fang Zhao, Lin Xiong et. al. Look Across Elapse: Disentangled Representation Learning and Photorealistic Cross-Age Face Synthesis for Age-Invariant Face Recognition Retrieved from https://arxiv.org/abs/1809.00338

[5] Alex D.Torres,HaoYan,Armin Haj Aboutalebi,Arun Das,Lide Duan, PaulRad.Chapter 3 - Patient Facial Emotion Recognition and Sentiment Analysis Using Secure Cloud With Hardware Acceleration. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780128133149000037

[6] Christopher Pramerdorfer, Martin Kampel.Facial Expression Recognition using Convolutional Neural Networks: State of the Art. Retrieved from https://arxiv.org/abs/1612.02903

[7] Omkar M. Parkhi, Andrea Vedaldi and Andrew Zisserman.Deep Face Recognition. Retrieved from http://www.bmva.org/bmvc/2015/papers/paper041/index.html

[8] Florian Schroff, Dmitry Kalenichenko, James Philbin.FaceNet: A Unified Embedding for Face Recognition and Clustering. Retrieved from https://arxiv.org/pdf/1503.03832.pdf

[9] Referenced from previous work https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/notebooks

[10] 5 Celebrity Faces Dataset. Retrieved from https://www.kaggle.com/dansbecker/5-celebrity-faces-dataset

[11] FacialExpressionRecognition2013Dataset.Retrieved from https://www.kaggle.com/msambare/fer2013

[12] Facenet json model https://github.com/serengil/tensorflow-101/blob/master/model/facenet_model.json

[13] Github link to our code implementation https://github.com/aishwaryakore5696/Face-Recognition-and-Emotion-recognition-for-Sentiment-Analysis