Project Report  
on

**DEEP LEARNING BASED FACE RECOGNITION**

*Submitted in the partial fulfilment of the requirements for  
the award of the degree of*

**BACHELOR OF TECHNOLOGY**

In

**ELECTRONICS AND COMMUNICATION ENGINEERING**

By

**K. PRAVEEN KUMAR 16311A04D8**

**K. VINAY KUMAR 16311A04D9**

**MD. ABU SUFIAN 16311A04E7**

Under the Guidance of

**Dr. Shruti Bhargava Choubey**

Associate Professor

Department of E.C.E



**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Autonomous)**

**Yamnampet (V), Ghatkesar (M), Hyderabad – 501 301.**

**April 2020**

**SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY   
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)  
Yamnampet (V), Ghatkesar (M), Hyderabad – 501 301**

**Department of Electronics and Communication Engineering**

**CERTIFICATE**

This is to certify that the project report entitled ***“DEEP LEARNING BASED FACIAL RECOGNITION”*** is being submitted by

**K. PRAVEEN KUMAR 16311A04D8**

**K. VINAY KUMAR 16311A04D9**

**MD. ABU SUFIAN 16311A04E7**

in partial fulfilment of the requirements for the award of **Bachelor of Technology** degree in **Electronics and Communication Engineering** to **Sreenidhi Institute of Science and Technology** affiliated to **Jawaharlal Nehru Technological University, Hyderabad** (Telangana)**.** This record is a bona fide work carried out by them under our guidance and supervision. The results embodied in the report have not been submitted to any other University or Institution for the award of any degree or diploma.

**Name of internal guide Name of Project Coordinator**

**Dr. Shruti Bhargava Choubey Dr. D. AJITHA**

Associate Professor Associate Professor

Department of E.C.E Department of E.C.E

**Head of the Department**Dr. S.P.V. SUBBA RAO  
Professor, Department of E.C.E

Signature of the External Examiner

**DECLARATION**

We hereby declare that the work described in this thesis titled **“*DEEP LEARNING BASED FACIAL RECOGNITION*”** which is being submitted by us in partial fulfilment for the award of Bachelor of Technology in the Department of **Electronics and Communication Engineering,** Sreenidhi Institute Of Science and Technology is the result of investigations carried out by us under the guidance of **Dr. Shruti Bhargava Choubey, Associate Professor**,  **Department of E.C.E, Sreenidhi Institute of Science and Technology, Hyderabad.**

No part of the thesis is copied from books/ journals/ internet and whenever the portion is taken, the same has been duly referred. The report is based on the project work done entirely by us and not copied from any other source. The work is original and has not been submitted for any Degree/Diploma of this or any other university.

Place: Hyderabad

Date:

**K.PRAVEEN KUMAR 16311A04D8**

**K.VINAY KUMAR 16311A04D9**

**MD.ABU SUFIAN 16311A04E7**

**ACKNOWLEDGEMENT**

We would like to thank my supervisor **Dr.Shruti Bhargava Choubey, Associate Professor, Department of E.C.E** for giving us constant guidance, support and motivation throughout the period this course work was carried out. Their readiness for consultation at all times, their educative comments and assistance even with practical things have been invaluable. We are thankful that she had gave the freedom to do the work with our ideas.

We would like to thank our project coordinator **Dr. D. AJITHA,** associateprofessor, Department of ECE for giving us constant guidance, support and motivation throughout the period this course work was carried out.

We express my sincere gratitude to **Dr. S.P.V. SUBBA RAO**, Head of Department, E.C.E for helping us in carrying out this project giving support throughout the period of our study in SNIST.

We are also thankful to our principle **Dr.CH. SIVA REDDY**, giving us this guidance and support, motivation throughout the period of our B.Tech course work carried out.

We convey our special thanks to honorable Executive Director **Dr.P.N. REDDY SNIST** for his continuous support in maintaining pleasant and friendly atmosphere in campus for studies and course work.

We are also thankful to all teaching and non-teaching staff of our department who has rendered their co-operation in completion of this project report.

We also thank our parents and friends and well-wishers who aided us in completion of the project report.

**K.PRAVEEN KUMAR 16311A04D8**

**K.VINAY KUMAR 16311A04D9**

**MD.ABU SUFIAN 16311A04E7**

**ABSTRACT**

Deep learning, specifically “Convolutional Neural Network(CNN)”, have accomplished success outcomes in face recognition as of late. In any case, it stays a open inquiry: why CNN’s function admirably and how to plan the 'decent' engineering. Present examinations will in general spotlight on revealing CNN designs that function admirably for face recognition as opposed to research the explanation. In this work, we lead a broad assessment of CNN based face recognition frameworks CNN FRS on the shared view to complete the work effectively reproduced. In particular, they utilize open database LFW “Labeled Faces in Wild” to prepare CNN’s, dissimilar to the most present CNN’s prepared on the private databases. We have proposed three CNN models which were the principal detailed designs prepared utilizing LFW information. The report significantly looks at a structures of the CNN’s and assesses an impact of various usage decisions. We distinguish a few helpful characteristics of CNNFRS. Take example, the depth of’ the scholarly highlights could be fundamentally diminished without advertisement stanza impact on the face acknowledgement precision. Likewise, the customary measurement knowing strategy misusing CNN-learned highlights is assessed. Examinations demonstrate two pivotal variables to great “CNNFRS” execution are a combination of various CNN’s and machine learning.

**Contents**

Acknowledgments

Abstract

**CHAPTER 1 INTRODUCTION Page No.**

1.1Introduction 11

1.2 Brief History 13

1.3 Problem statement 14

1.4 Motivation 14

1.5 Related work 15

**CHAPTER 2 LITERATURE SURVEY**

2.1 Face Detection and Face Tracking 17

2.2 Face Positioning and Alignment 17

2.3 Face Feature Extraction 18

2.4 Theoretical idea of Proposed work 19

**CHAPTER 3 BUILDING FACE RECOGNATION MODEL WITH NEURAL NETWORK**

3.1 Introduction to Neural Network 20

3.2 Convolution Neural Network 21

3.2.1 Build Face Recognition Model with CNN 22

3.2.2 Theory 22

3.2.3 Build Siamese Network with CNN 24

**CHAPTER 4 FULLY LINKED NEURAL NETWORKS**

4.1 What Is a Fully Connected Deep Network? 26

4.2 AI Winters 30

4.3 Universal Convergence Theorem 31

4.4 Flow Chart 33

**CHAPTER 5 DEVELOPING STRONG FACE RECOGNITION SYSTEM**

5.1 Introduction 34

5.2 Triplet Loss 35

5.3 Deep Convolutional Networks 37

5.4 Datasets and Evaluation 39

5.5 Datasets for Academic Purpose 41

**CHAPTER 6 RESULTS AND DISCUSSIONS**

6.1 Computation Accuracy Trade off 42

6.2 Upshot of CNN Model 44

6.3 Affectability of Picture Quality 45

6.4 Embedding Dimensionality 46

6.4.1 Extent of Training Data 46

6.4.2 Performance on LFW 48

6.4.3 Performance on YouTube Faces Database 48

6.4.4 Face Clustering 48

**CHAPTER 7 CONCLUSIONS & FUTURESCOPE**

7.1 Conclusion 50

7.2 Future Scope 50

**References**

**Appendix** 52

**Plagiarism check** 73

**Conference Paper** 76

**LIST OF FIGURES**

**S. No. Figure No. Figure Name Page No.**

1 Figure 3.1 Convolution Neural Network 21

2 Figure 3.2 Layers in CNN 23

3 Figure 3.3 Siamese Network Work Flow 25

4 Figure 4.1 Fully connected network 27

5 Figure 4.2 A Multilayer connected network 28

6 Figure 4.3 Representation of Neuron 29

7 Figure 4.4 Representation of Perceptron 30

8 Figure 4.5 Perceptron’s linear rule 31

9 Figure 5.1 CNN Architecture 35

10 Figure 5.2 CNN 36

11 Figure 6.1 Flops compared with accuracy tradeoff 42

12 Figure 6.2 Architecture of Network 44

**LIST OF TABLES**

**S.No. Table No. Table Name Page No.**

**1** Table 5.1 Accuracy of facial recognition models 34

**2** Table 5.2 Zeiler & Fergus model NN1 38

**3** Table 5.3 NN2 40

**4** Table 6.1 The Architecture of networks 42

**5** Table 6.2 Quality of Image 45

**6** Table 6.3 Embedding Dimensionality 46

**7** Table 6.4Data size of Training data 47

**Abbreviations:**

CNN Convolutional Neural Network

LFW Labeled Faces in Wild

GPU Graphical Processing Unit

YTF YouTube Faces

LBP Local Binary Patterns

SIFT Scale Invariation Feature Transform

HOG Histogram of Oriented Gradients

LGBPHS Local Gabor Binary Pattern Histogram Sequence

AI Artificial Intelligence

## 

## **CHAPTER 1**

## **INTRODUCTION**

## **1.1 INTRODUCTION**

Human being facearecognition is the difficult issue in PC vision.with a few biometric’s application’s. The issue basically face’s challenges because of varieties in facial appearance brought about by elements, for example, brightening, demeanor, and fractional impediment from embellishments including glasses, scarves, caps, and such.

As of late, profound learning, based methodologies had been progressively applicable for human’s face recognition for promising outcomes. These strategies accept crude information as their system include and convolve channels in various levels to naturally find low level and elevated level representations from named, or not labeled information for recognizing, recognizing, as well as characterizing their fundamental examples. Be that as it may, streamlining a huge number of parameters to take in the multi-stage loads without any preparation in profound learning designs requires a huge number of preparing tests and an entrance to amazing computational assets, for example, Graphical Processing Units (GPUs). Thusly, the strategy for move learning is proficiently used to apply recently learned information on a pertinent visual recognition issue to the new, wanted assignment space.

Move learning can be applied in two unique manners as for the structure and closeness among the prepreparing dataset. The primary methodology is tweaking the pre-prepared system loads utilizing the new dataset by means of backpropagation. This strategy is just recommended for huge enough datasets since adjusting the preprepared systems with barely any preparation tests can prompt overfitting. The subsequent methodology is the immediate usage of scholarly loads in the ideal issue to separate and later characterize highlights. This plan is particularly proficient when a new data set is little and additionally a couple of no of classe’s exist. Contingent upon a errand comparability among the two datafiles, we can conclude should we utilize bring down layers loads as a nonexclusive low level component extractors or high layers loads as assignment explicit theme extractors .

In our report, the top layer part of took in loads from the two profound “convolutionalaneuralanetworks”(CNN’s) of “VGGFace” and Lightened CNN preprepared on enormous face detection assortments, had been utilized to remove the face portrayal.They have picked these two modess because they are fruitful.The previous system incorporates the profound design and next is very efficient CNN. Power of the profound face potrayal’s against varieties of various variables includes enlightenment, impediment, posture, and the mis-alignment had been altogether surveyed utilizing the five mainstream face datafiles, in particular “AR,CMUPIE”,”Extended Yale datafile”,ColorFERET, and FRGC.

The principle commitments ,results of our work can be abridged as shown below:

(i)An extensive assessment of Deepalearning based portrayal into different conditions which includes present, light, impediment, misalignment have been led. Actually, all the proposed deepalearning based face recognition techniques, for example, “Deep-Face, Deep-ID , Face-Net” , and VGGFace had been prepared and they assessed on exceptionally enormous animal face recognition datafiles, for example Named Face’s in a Wild (LFW) , YouTube Faces (YTF),and Mega-Face. Be that as it may, their portrayal capacities to deal with singular appearance varieties have not been surveyed at this point.

(ii) We have indicated that albeit profound learning gives an incredible portrayal to confront recognition, it can't accomplish cutting edge results against posture, light, and impediment. To empower profound learning models accomplish better outcomes, either these varieties ought to be considered during preparing or preprocessing strategies for posture and enlightenment standardization ought to be utilized alongside pre-prepared model.

(iii) Weahave discovered that the profound learning, based face portrayal is powerful to mis-alignment and ready to endure facial element confinement blunders approximately to 10% of an interocular separation.

(iv) The “VGGFace” model is transferrable if we compare with the CNN model. Overall I believe that more research has to be done in face recognition because under few conditions there is a mismatch in order to rectify that research has to be done.

In this paper section 2 contains the review of the previous existing recognition methods. In part 3 there is a detailed version of two deep CNN methods and how to approach for the recognition of face based on the above models. In section 4 we have used our datasets for face detection and we have presented our results based on that. In the final section we have concluded our discussion and what are the demerits and in future how we can overcome our disadvantages.

## **1.2 BRIEF HISTORY:**

In the year of 1964,1965 “Bledsoe,Helen-Chan” and “Charles Bisson”, have worked together to detect the faces in computer. They were very proud of their work and because of their insufficient money and during their time the publications were done by an intelligence agency which haven’t allowed publicity they haven’t published their entire work. Based on few available resourses they have found that the Bledsoe’s approach could detect the eyes,mouth and could detect the face in different angles and in different conditions and is compared with the images to detect the person.

In this he have given a large set of database containing pictures and a photo. And the question here is that the computer has to select one of the picture or image from the database which matches with the photograph. He have measured his success based on the no of images present in the database. If there are more number of images then his success rate is high.

The project was named as a man-machine project because we have used the coordinate’s from the photo’s and they were used by the computer for the purpose of recognition. By using graphics tablet we can extract the feature’s like centre of the retina,the corners of the eyes present inside and outside and the point of cornea and many more features. From the extracted coordinated few calculations were done like width of the lips,mouth and the width of the eyes and retina to pupil distance. This operators process nearly 40 pictures per one hour. When we are constructing the database the name of the man present in the photo will be associated with the computed distances and the it is stored in the computer memory. In the phase of recognition the distances were compared with the corresponding one for each photo after finding the distance between them the closest match is returned.

## **1.3 PROBLEM STATEMENT:**

## The problem of face recognition is as follows: we will give a input image and there will be few images already recorded in the database. Based on the images already present we have the detect the input image. For example we give Bill gates picture as an input image then already Bill gates pictures should be present in the database based on that we will detect the image of Bill gates.

## **1.4 MOTIVATION:**

In recent days we are mostly using the biometric techniques for the detection of the persons instead of recognizing the people based on the passwords, PIN’s, smart and plastic cards, keys and tokens and many more this strategies will examine the persons “physiological” and “behavioral” characteristics in order to determine the person. We can easily forget the passwords and also pins and there can be stolen also or some members will guess our passwords easily. Cards and tokens and keys can be forgotten or misplaced or purloined or they can be duplicated coming to magnetic cards they can be corrupted and also they are unreadable however our face cannot be stolen or misplaced or duplicated or forgotten or forged.

Face recognition have many merits when compared with the biometric technology few are quoted below: almost in all the technologies there is something that should be done by the user like in biometric technology the user have to place his hand on the machine and in Iris recognition we have st stand at a certain distance so that our retina images may be captured by the camera where as in face detection there is nothing that the user should do since the images of the face can be captured from some distance itself. This is beneficial regarding the security purposes. Further there were many fraught done in the biometric technology if the tissue epidermis which is present on the finger is damaged then it is difficult to detect. Coming to Iris detection we require much expensive equipment’s and are sensitive too. Where as in voice recognition there is a demerit of background noise etc. In regard to the above factors we can say that face detection is easy and less expensive and most accurate method

## 

## **1.5 RELATED WORK:**

Prior to the rise of profound learning calculations, most of conventional facearecognition techniques have first extracted the hand created shallowed highlightsafrom facial pictures utilizing “LocalBinaryPatterns” (LBP), “Scale InvariationaFeatureaTransform” (SIFT), and“Histogram of OrientedGradients” (HOG),. Be that as it may, with the benefit capacity of the cutting edge computational assets andawith a flood inorder to access to huge datafiles, “Deep learning” models had been created to demonstrate gigantically great outcomes for various visual recognition errands including face recognition .

DeepFace isaone among the exceptional systems which have a nine layered profound CNN-model with the two convolutional layer’s and more of 120M parameter’s preparedaon 4,000,000 facial pictures from more than 4,000 characters. This technique, through arrangement of pictures dependent can achieve the accuracy of 97.3% and 97.45% on “LFW” .

Facenetais a profound “CNN” which dependens on Google Net and system is proposed and prepared based on the face datafile with in between100 and 200M pictures around 8,000,000 characters. This calculation utilizes triplets of generally adjusted countenances got from and legitimately figures out how to delineate pictures to a com-agreement “Euclidean space” to gauge the face closeness.

# 

## **CHAPTER 2**

## **LITERATURE SURVEY**

## **2.1 FACE DETECTION AND FACE TRACKING**

The article named “Robust Realtime Object Detection” is very most every now and again refered to articleainaa progression of article’s by “Viola” which makes face discovery genuinely useful. We can find out around a few face discovery techniques and calculation’s from the above distribution. The article named “Fast revolution in-variant multisee face identification” dependen’s on the genuine adaboost just because genuine adaboost applied to question recognition, and proposed an increasingly develop and functionalamultiface location ,home structured referenced on a course structure upgrades likewise have great outcomes.

More than 3 paper’s have examined about a face location and the face tracking issues. As indicated by the exploration bring about these papers, we can make constant face identification frameworks. The important feature is to detect size and the position of face in the vedio or image but in regard to tracking it is important to determine the similarity between the faces in the casing.

## **2.2 FACE POSITIONING AND ALIGNMENT**

Prior restriction of the facial component focuses on a few keyapoints, for example, finding the focal point of the eye and the middle point of our mouth, yet next presented much more focuses and then added shared limitation in order to correct the exactness,security of situating Sex. Theaarticle named “ActiveaShapeaModels Their Trainingaand Application” is the model of many face component which focuses, surface andapositional relationship imperatives thought about togetherafor computation. Though ASM has many more aricles one more important idea is to improve the article which is based on the texture concept. The approach which is based on the regression is better than the categorical model. This article is the improvement of shape model.The reason for the facial component point situating is to additionally decide facial element focuses (eyes, mouth focus focuses, eyes, mouth shape focuses, organ form focuses, and so forth.) based on the face territory distinguished by a face recognition/following, position’s. Those are the articles which focuses more on face positioning/alignment. The main idea to find the face features is to join the texture features ,constarints.

## **2.3 FACE FEATURE EXTRACTION**

Eigen faces which are based on PCA are one among the great calculation’s to detect the face. Albeit the present PCAais progressively utilized in the reduction of the dimensions in real wolrd than classification like the classic approach. The article name “LocalaGaboraBinaryaPattern Histogram Sequence” (LGBPHS): In numerous down to earth frameworks, a system for removing validation data is the PCA and the LDA. Utilizing PDAato lessen frameworkto evade theagrid peculiarity issue of LDA explaining, at that point utilizing LDA to extricate the highlights reasonable for grouping, To additionally recognize the different unique highlights removed after the choice level combination. Albeit a portion of the “LFW” test conventions areanot sensible.

The papers above have discussed regarding the positioning and aliagnment of face. Human’s face to make reference to an attribute of the procedureaof information "a face guide" , "facial highlights key focuses facilitates", which yield is the comparing the face with that of the string (include). Up close and personal component calculation willabe founded on the facial highlights of a keyApoint directions of a human’s face predecided mode and afterward ascertain theahighlights. Lately, the profound learning technique essentially controlled the cosmetic touch up highlight calculation, In the above mentioned articles, they have demonstrated the advancement of the research around. The calculations are fixed length calculations. Prior technology include models which are bigger, very slow, just utilized out of sight administration. Some studies have decreased the size and the speed of the operation time.

## **2.4 THEORETICAL IDEA OF PROPOSED WORK**

Face detection is similar to that of the pattern recognition which converts the image into numbers which is understood by the computers. If we have a picture of 256 bits,each and every pixel of the picture will be lying between 0 to 255 by this we can convert the picture into the matrix/lattice. How to distinguish the examples in this network? One path is to utilize a moderately little grid to clear from left to right and through and through in this enormous framework. Inside every little lattice square, we can check the quantity of events of each shading from 0 to 255. So we can communicate the attributes of this square.

Through this sweep, we get another grid comprising of numerous little lattice square highlights. Also, this grid is littler than the first network. At that point, for this littler grid, play out the above advances again to play out a component "focus". In another sense, it is preoccupied. At long last, after numerous deliberations, we will transform the first network into the one measurement by one measurement grid, is the number. Different type of pictures is abstracted into different numbers. Next calculate the difference between the matrices then we can compare the faces.

# **CHAPTER 3**

# **BUILDINGAFACE RECOGNATION MODELAWITH NEURAL NETWORK**

## **3.1 INTRODUCTION TO NEURAL NETWORK**

From the year of 1980 “Artificial neural n/w” is the main in the artificial intelligence. It takes the human cerebrum neuron’s and based on the different connection method’s it forms different models and different networks. It is likewise frequently alluded as neural system or neural network inabuilding and the scholarly community. Every hub speaks to a particular yield work called an initiation work. The association between each two hubs speaks to a weighting an incentive to pass the association signals, called the weight, it is equal to the memory of the counterfeit neural systems. Yield of a system differs relying upon the association technique for the system, the weight esteem and the excitation work. The system itself is typically an estimation of a calculation or capacity in nature.

From the decade, there was dramatical increase in the research of artificial networks. Many problems like pattern detection, economy, medicine, predictive estimation, intelligent robot’s, biology have been solved successfully. The problems which are practical in nature which are difficult show’s very good intelligence.

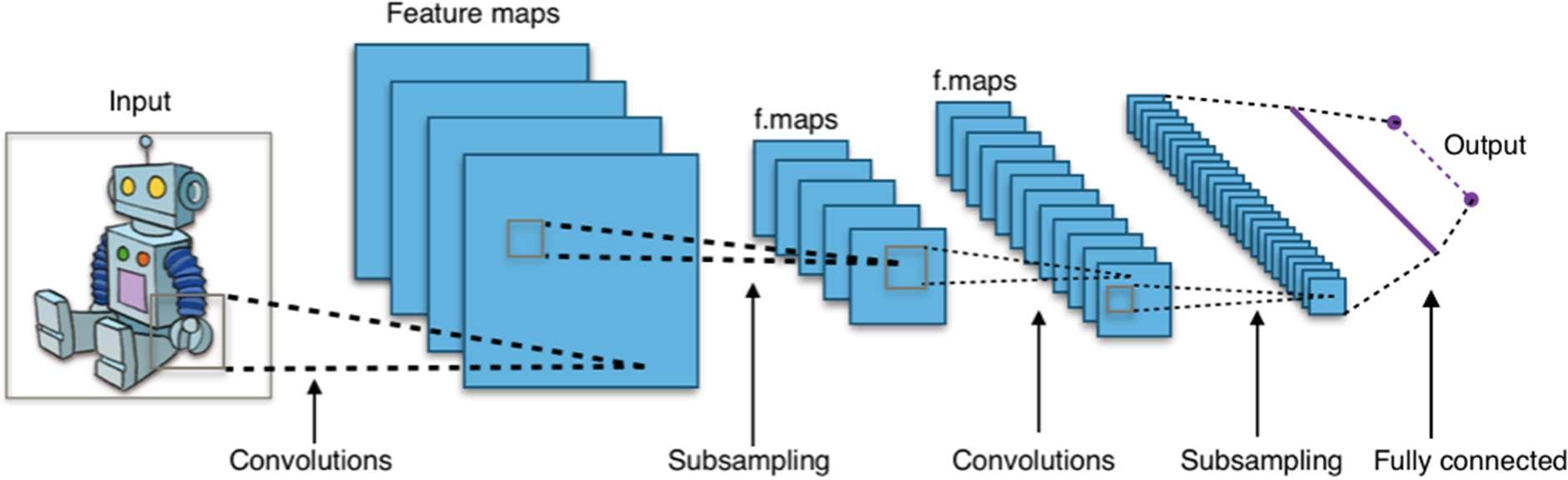
The neuron characteristics and the learning rules and the network topology are mainly considered by the artificial networks .As of now there are approximately 40 types of neural models,perception,Boltzmann machine, Hopfield netwoks etc.According to network topology of connection there are two types of network models like Feed back network and feedforward network.

Feedforward system: Each and every neuron takes the input from the previous state and gives that output as a input to the next state. As there is no feedback in the system it is represented by the free flow graph. This system knows how to transform a signal from one state to another and it has the ability to process information from the recombination of many nonlinear functions. This system is simple and implementation is also easy.

Feed back system: In this system the part of the output is given back to the input between the nerves/neurons there are shown by a undirected complete graph. The theory of dynamic system takes care of the information processing of the neurons. The stability comes in closure to the memory function which is associative in nature Boltzmann machine and the Hopfield systems comes under this category.

## **3.2 CONVOLUTIONAL NEURAL NETWORK**

“Convolutional neural networks” (CNN) comes under the deep networks the are applied mostly for visualizing the images. This belong to the feed back system category. The main difference between CNN and the multiple layered perceptron is nothing but the network. In this first we will take a image of small size nothing but filter and we will move the small sized image on the large size image like this we will carry out the action in different places this is nothing but convolution .



3.1.convolution neural network

The filter size is 5\*5\*3 and the size of the image is 32\*32\*3. If we observe both we can find that the width of the filter and the image is same. Once the convolution is completed next we have to go for pooling which is the second step. In simple words pooling means reducing the size of the image. There are many types of poolings out of which we are going to select a pooling named max pooling. In this pooling the maximum size from each feature map is selected and from that we are going to build another feature map.

## 

## **3.2.1 BUILD FACE RECOGNITION MODEL WITH CNN**

## **We can divide the face recognition methods broadly into two types:**

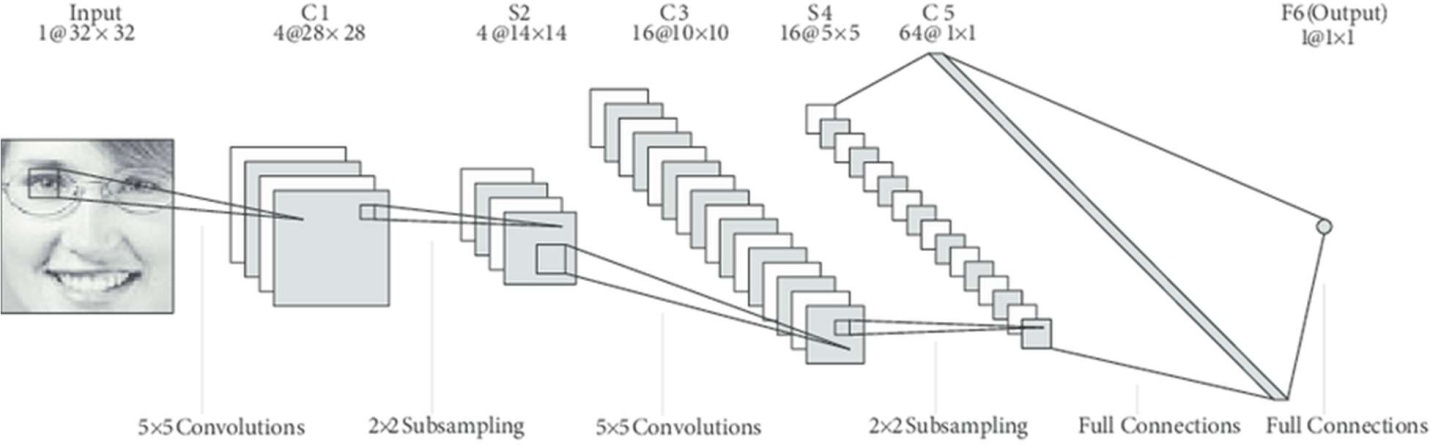
(1) “Representation based strategies:” The main and simple theme is to convert the image into another space and with the help of the statistical methods we can examine the face pattern

(2) “Feature based strategy:” In this strategy we extracts the features and we will send that features to a classifier for detecting the face,such as the recognitions which are based on HMM. CNN which is used of the recognition of face is considered as a method based on features. This method is generally different from the traditional method of extracting features. In this method we will perform the convolution layer by layer and next multiple layered non-linear mapping is performed. This method will reduce the layers used in the network and also the samples used.

**3.2.2 THEORY**

“Convolutional neural networks” (CNN) comes under the deep networks the are applied mostly for visualizing the images. This belong to the feed back system category. The main difference between CNN and the multiple layered perceptron is nothing but the network. In this first we will take a image of small size nothing but filter and we will move the small sized image on the large size image like this we will carry out the action in different places this is nothing but convolution .

The filter size is 5\*5\*3 and the size of the image is 32\*32\*3. If we observe both we can find that the width of the filter and the image is same. Once the convolution is completed next we have to go for pooling which is the second step. In simple words pooling means reducing the size of the image. There are many types of poolings out of which we are going to select a pooling named max pooling. In this pooling the maximum size from each feature map is selected and from that we are going to build another feature map.

3.2.layers in CNN

Convolution layer: In this layer by using the methods which are simple in nature like weight sharing and local connection we are extracting some of the local features. In local connection the neurons which are present in the convolution layer are connected with the neurons present in the fixed area in the past feature map.

weightasharing implies thatathe neurons in a similar element map utilize a similar association quality and the past layer. Association, can lessen the system preparing parameters, a similar arrangement of association quality is an element extractor, which is acknowledged as a convolution portion during the time spent computation, and the convolution piece esteem is arbitrarily introduced first, lastly controlled by organize preparing.

The pooling or sampling layer: The next layer after convolution is pooling layer the main aim of this is to reduce the size of the image. There are many poolings oout of which we are going to select the max pooling. After the sampling is done the no of features remains unchanged but the size of the feature map is reduced as we are selecting only one from each feature map which is maximum.

## **3.2.3 BUILD SIAMESE NETWORK WITH CNN**

In the wake of looking at changed neural systems and their qualities, we utilized Siamese system to determine the issue. The Siamese system is neural system for estimating of closeness. It tends to be utilized for classification distinguishing proof, characterization, and so forth., in the situation when there are numerous classifications he quality of test is very less. The customary arrangement technique for recognizing is to know the precisely which classes each example has a place with and need to have an accurate mark for each example. Also, the overall number of labels isn't excessively. These strategies are less material when the quantity of classifications is excessively enormous and the quantity of tests per classification is moderately little. Truth be told, it is likewise very surely known. For the whole informational index, our information volume is accessible, however for every class, there can be just a couple of tests, at that point utilizing the arrangement calculation to do it, in light of the fact that every classification of tests is excessively Less, we can't prepare any great outcomes whatsoever, so we can just locate another approach to prepare this informational index, hence introducing a Siamese system, shown in Figure3.3

The Siamese system will learn the similar measure and it will use that to compare and join the samples of new category. This method is used when the classes are large in nature and when the samples cannot be used for the previous one.

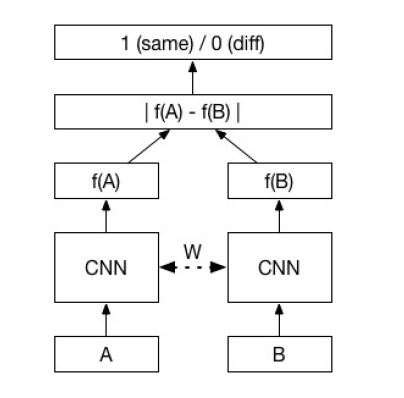


Fig 3.3.Siamese Network Work Flow

The Siamese system is neural system for estimating of closeness. It tends to be utilized for classification distinguishing proof, characterization, and so forth., in the situation when there are numerous classifications he quality of test is very less. The customary arrangement technlque for recognizing is to know the precisely which classes each example has a place with and need to have an accurate mark for each example. Also, the overall number of labels isn't excessively. These strategies are less material when the quantity of classifications is excessively enormous and the quantity of tests per classification is moderately little. Truth be told, it is likewise very surely known.

## **CHAPTER 4**

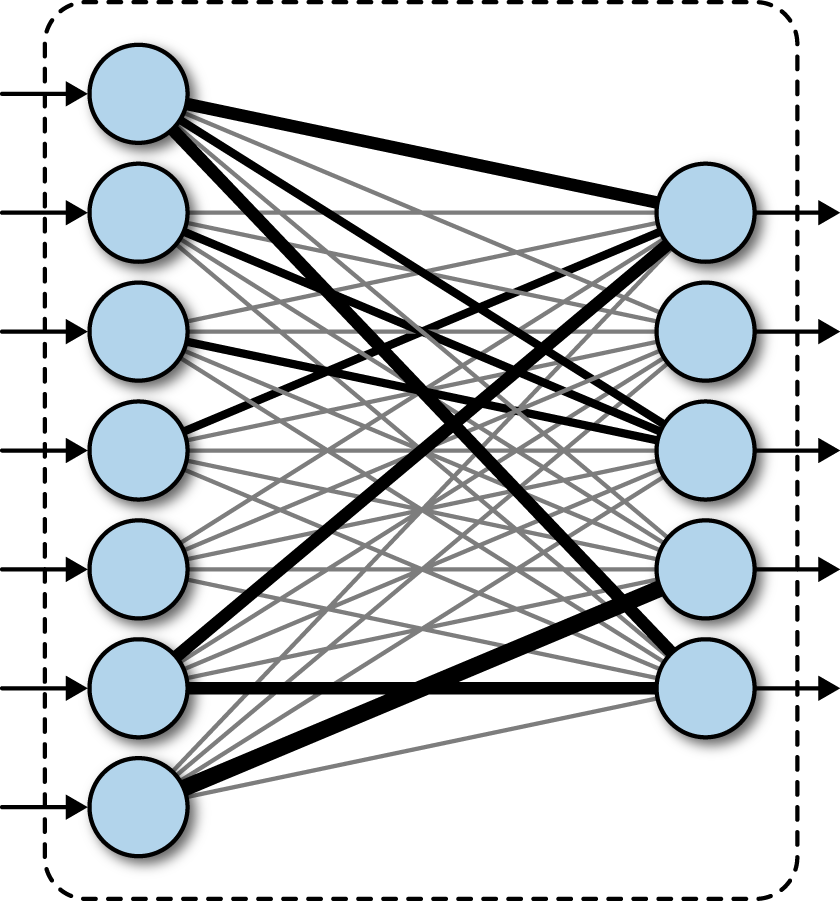
## **FULLY LINKED NEURAL NETWORKS**

This section will acquaint you with completely associated profound systems. Completely associated systems are the workhorses of profound learning, utilized for a large number of utilizations. The significant preferred position of completely associated systems is that they are "structure skeptic." That is, no extraordinary suspicions should be made about the contribution (for instance, that the info comprises of pictures or recordings). We will utilize this sweeping statement to utilize completely associated profound systems to address an issue in compound demonstrating later in this section.

We dig quickly into the numerical hypothesis supporting completely associated systems. Specifically, we investigate the idea that completely associated models are "widespread approximators" fit for learning any capacity. This idea clarifies the sweeping statement of completely associated structures, yet accompanies numerous provisos that we talk about at some profundity. While being structure freethinker makes completely associated arrangements comprehensively pertinent, such systems will, in general, have more fragile execution than unique reason systems tuned to the structure of an issue space. We will examine a portion of the impediments of completely associated designs later in this part.

**4.1 What Is a Fully Connected Deep Network?**

A completely associated neural system comprises of a progression of completely associated layers. A completely associated layer is a capacity from ℝ m to ℝ n. Each yield measurement relies on each information measurement. Pictorially, a completely associated layer is spoken to as follows in Figure 4.1.

 Figure 4.1. A fully connected layer in a deep network

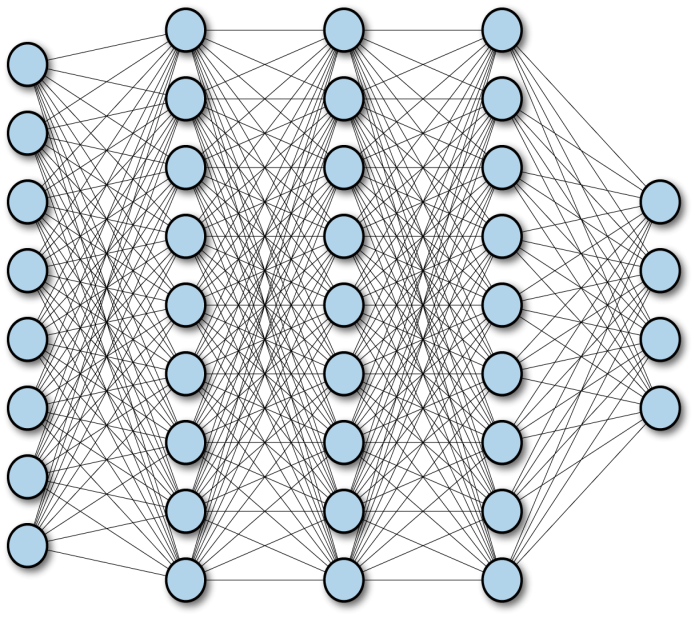
How about we delve somewhat more profound into what the numerical type of a completely associated arrange is. Let x ∈ ℝ m speak to the contribution to a completely associated layer. Let y I ∈ ℝ be the I-th yield from the completely associated layer. At that point y I ∈ ℝ is figured as follows:

y I = σ ( w 1 x 1 + ⋯ + w m x m )

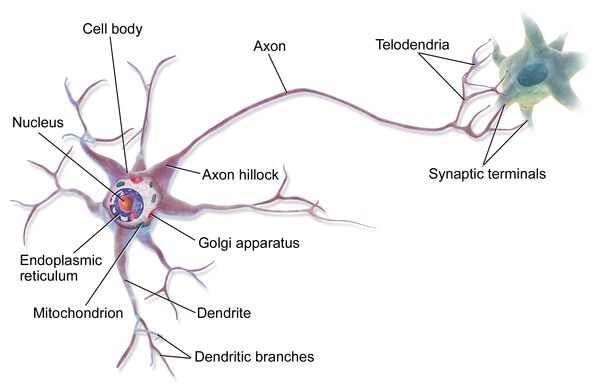
Here, σ is a nonlinear capacity (for the present, consider σ the sigmoid capacity presented in the past section), and the w I am learnable parameters in the system. The full yield y is at that point

y = σ ( w 1,1 x 1 + ⋯ + w 1,m x m ) ⋮ σ ( w n,1 x 1 + ⋯ + w n,m x m )

Note that it's straightforwardly conceivable to stack completely associated systems. A system with various completely associated systems is frequently called a "profound" organize as portrayed in Figure 4-2.



###### Figure 4.2. A multilayer deep fully connected network.

As a brisk usage note, note that the condition for a solitary somatic cell appearance an equivalent as a dab result of 2 vectors (review the voice communication of tensor rudiments). For a layer of neurons, it's oft useful for productivity functions to work y as a grid increase:   
y=σ(wx )   
where the letter of the alphabet may be a framework in ℝ n×m and therefore the nonlinearity σ is applied componentwise.   
"Neurons" in absolutely Connected Networks: The hubs in utterly associated systems are commonly alluded to as "neurons." Consequently, in different places within the writing, utterly associated systems can usually be alluded to as "neural systems." This classification is to a good extent Associate in Nursing authentic mishap.   
During the Forties, Warren S. McCulloch and music director Pitts distributed a primary numerical model of the mind that contended that neurons were equipped for reckoning subjective capacities on Boolean amounts. Successors to the current work somewhat refined this consistent model by creating scientific "neurons" nonstop capacities that modified somewhere within the vary of zero and one. On the off likelihood that the contributions of those capacities developed huge enough, the somatic cell "terminated" (took on the value one), else was peaceful. With the growth of movable hundreds, this portrayal coordinates the past conditions.   
  
Is this, however, a real somatic cell carries on? not! a real neuron (Figure 4-3) is Associate in Nursing passing puzzling motor, with quite a hundred trillion molecules, and a large range of varied tired proteins equipped for reacting to differing signals. A chip may be a superior similarity for a somatic cell than a one-line condition.

###### Figure 4.3. A more biologically accurate representation of a neuron.

From multiple points of view, this distinction between natural neurons and fake neurons is very deplorable. Unenlightened specialists read short of breath official statements guaranteeing counterfeit neural systems with billions of "neurons" have been made (while the mind has just 100 billion natural neurons) and sensibly leave away accepting researchers are near making human-level insights. Cutting edge in profound learning is decades (or hundreds of years) away from such an accomplishment.

As you read further about profound learning, you may run over overhyped guarantees about computerized reasoning. Try not to be hesitant to get out these announcements. Profound learning in its present structure is a lot of procedures for tackling analytics issues on quick equipment. It's anything but an antecedent to Terminator.

# **4.2 AI WINTERS**

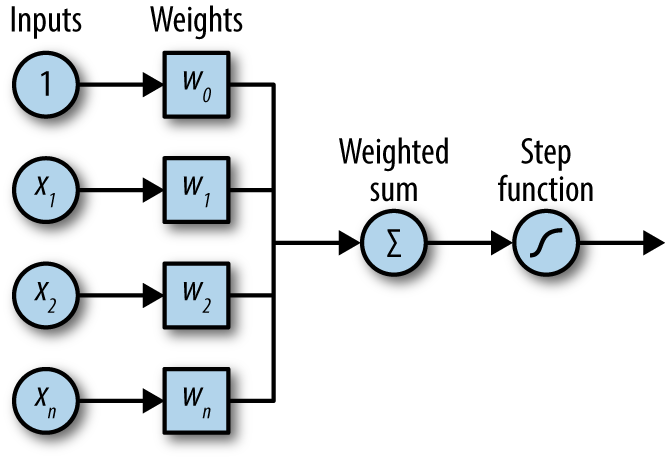
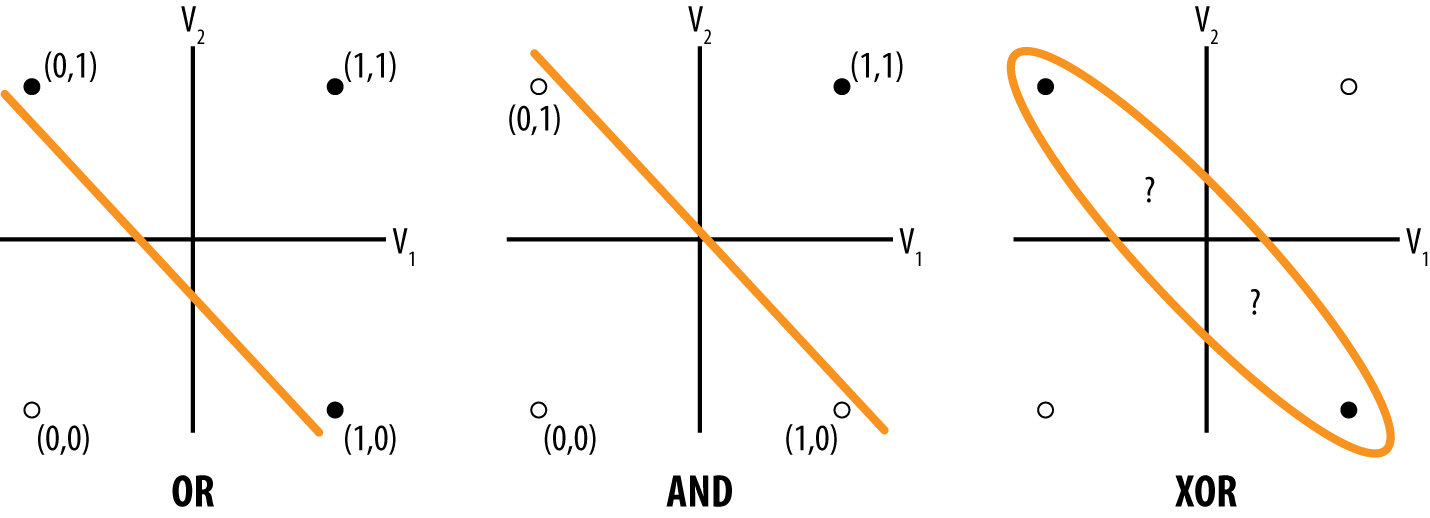
Artificial Intelligence has knowledgeable different rounds of blast and-bust flip of events. This repetition improvement is traditional for the sector. every new development in learning produces a flood of fine religion whereby prophets guarantee that human-level (or superhuman) insights are early. Following a few of years, no such insights show, and pissed off funders pull out. the following amount is named AN AI winter.   
There are various AI winters up till this time. As a thought total, we tend to urge you to contemplate once the subsequent AI winter can occur. this inflow of profound learning progress has tackled heaps additional cheap problems than any past rush of advances. Is its conceivable AI having at long last embarked on and left the blast and-bust cycle or does one believe we're sure the "Incomparable Depression" of AI soon?   
Adapting Connected Networks with Backpropagation   
The principal adaptation of a very associated neural system was the Perceptron, (4.4), created by Frank Rosenblatt throughout the Fifties. These perceptrons are indistinguishable from the "neurons" we have a tendency to given within the past conditions.

Figure 4.4. A diagrammatic representation of the perceptron.

Perceptrons were ready by a custom "perceptron" rule. whereas they were creditably valuable taking care of simple problems, perceptrons were in a very general sense restricted. The book Perceptrons by Marvin Minsky and queen Papert from the end of the Nineteen Sixties incontestable that easy perceptrons were unequipped for learning the XOR work. Figure 4-6 outlines the verification of this announcement

###### Figure 4.5. The perceptron’s linear rule can’t learn the perceptron.

This issue was overwhelmed with the development of the multilayer perceptron (another name for a profound completely associated arrangement). This development was an impressive accomplishment since prior basic learning calculations couldn't learn profound systems adequately. The "credit task" issue puzzled them; how does a calculation choose which neuron realizes what?

The total answer for this issue that requires a propagation in backward. Backpropagation is a summed up necessity to learn loads of the neural system. Surprisingly, convoluted clarifications are a scourge in the writing.

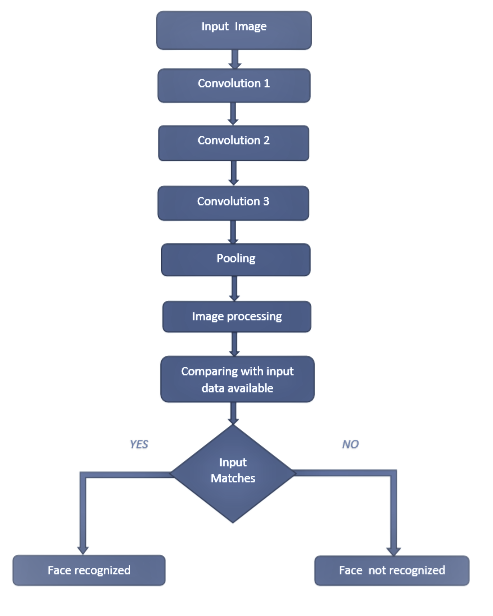
How about we guess that “f ( θ, x )” is a capacity that speaks to a profound completely associated arrangement. Here x is the contributions to the completely associated system and θ is the learnable loads. At that point the backpropagation calculation just processes ∂f ∂θ . The commonsense complexities emerge in actualizing backpropagation for every single imaginable capacity f that emerge practically speaking. Fortunately for us, TensorFlow deals with this as of now!

## **4.3 UNIVERSAL CONVERGENCE THEOREM**

The first conversation has addressed the thoughts that profound completely associated systems are amazing estimations. McCulloch and Pitts indicated that coherent systems can code (nearly) any Boolean capacity. “Rosenblatt's Perceptron” was the ceaseless simple of “McCulloch and Pitt's legitimate capacities”, however, it was demonstrated to be on a very basic level restricted by “Minsky and Papert”. Multilayer perceptron hoped to settle the confinements of straightforward perceptron and exactly appeared to be equipped for learning complex capacities. Notwithstanding, it wasn't hypothetically evident whether this observational capacity had unfamiliar confinements. In 1989, “George Cybenko” showed that a multilayer perceptron was fit for speaking to self-assertive capacities. This exhibition gave an extensive lift to the cases of consensus for completely associated organizes as a learning engineering, halfway clarifying they've proceeded with ubiquity.

In any case, if both backpropagation and completely associated organize hypothesis were comprehended in the late 1980s, for what reason didn't "profound" realizing become progressively mainstream before? A huge piece of this disappointment was because of computational impediments; adapting completely associated systems took an extreme measure of registering power. What's more, profound systems were exceptionally hard to prepare because of the absence of comprehension about great hyperparameters. Subsequently, elective learning calculations, for example, SVMs that had lower computational prerequisites turned out to be progressively famous. The ongoing flood in notoriety in profound learning is somewhat because of the expanded accessibility of better-registering equipment that empowers quicker figuring, and halfway because of expanded comprehension of good preparing regimens that empower stable learning.

**4.4 FLOW CHART:**



## **CHAPTER-5**

## **DEVELOPING STRONG FACE RECOGNITION SYSTEM**

**5.1 INTRODUCTION**

Best exactness face recognition models were accounted for within logical investigates by “goliath Technology organizations” and “research establishments”, as appeared in Figure. Be that as it may, all these historic outcomes despite everything remains in the research center. Uses of face recognition, in reality, are difficult to look. As an example, you cannot see it be utilized in DMV to void phony id, can't see it in a childcare agency to ensure the correct individual gets the correct child, you can't see it in a wellness club to make client registration progressively lovely.

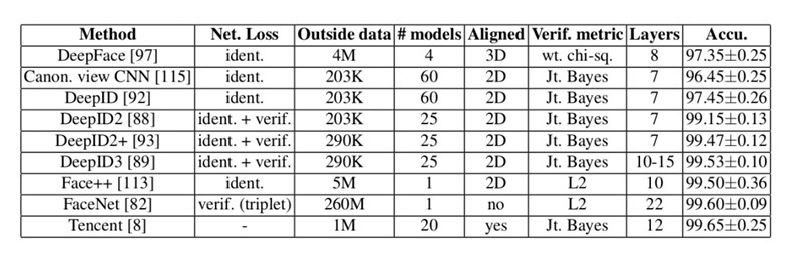


Table 5.1. Accuracy of facial recognition models

The keyhole between face recognition inquires about and modern utilization is the application. You can see distinctive calculation organizations gave diverse APIs, yet how to transform the API into the genuine item is an extreme issue laying before potential modern clients.

## 

## **5.2 TRIPLET LOSS**

The triplet-primarily based misfortune work used to emerge as familiar with the mapping is an adjustment of Kilian Weinberger's Large Margin Nearest Neighbour (LMNN) classifier (which greater than as soon as arranges images of a similar character and at the same time pushes pix of any great character away) to profound neural systems.

Sun et al. use groups of systems prepared to utilize a blend of characterization and check misfortune. The check misfortune they use is like the triplet misfortune used to get familiar with the mapping utilized by FaceNet in that it limits squared L2 removes between pictures of countenances from a similar individual and authorizes an edge isolating pictures of appearances from an alternate individual, however, it's distinctive in that lone sets of pictures are thought about, while the triplet misfortune supports a relative separation requirement by taking a gander at three at once.

A misfortune like FaceNet's triple misfortune was utilized by Wang et al. for positioning pictures by semantic and visual comparability. Regarding the CNN design as a BlackBox, the most significant piece of FaceNet lies at long last to-end learning of the framework.

Fig.5.1.CNN architecture

FaceNet searches for putting in f(x) from a picture into highlight area Rd, with the end intention that the squared L separation between all face photographs (self-sufficient of imaging states) of a similar character is little, while the separation between a couple of face snapshots from diverse characters is huge.

Though recently utilized misfortunes energize all countenances of a similar personality onto a solitary point in Rd, the triplet misfortune moreover attempts to uphold an edge between each pair of appearances from one individual (grapple and constructive) to every one of others' countenances. This edge implements discriminability to different characters.

5.2.CNN architecture

We need to guarantee that a picture x an of a particular individual is closer to every unmarried other image x p of that equivalent character than it's far to any picture x n of some other character by a facet α. That is, Along these lines, the misfortune (L) is: a = 0.2

Of every single imaginable triplet (N of them), many would effectively fulfill the above imperative. So it'd be a loss to take a gander at these during preparing (wouldn't add to modifying parameters, would just stoppage assembly); it's accordingly imperative to choose "hard" triplets (which would add to improving the model) to use in preparing. How would we do that?

**Triplet Selection:** A thought: Given a stay picture x a, select the "hardest" constructive picture (of a similar individual) as x p (for example the one that is uttermost away in the dataset) and select the "hardest" antagonistic picture (of an alternate individual) as x n (for example the one that is nearest in the dataset). If this triplet doesn't abuse condition, at that point none with that stay will. (Think: on the off chance that d- - d+ > a, at that point the condition is met.)

Issue: Infeasible to register these argmax and argmin over the entire dataset. Likewise, this may prompt poor preparation (taking into account that mislabelled and inadequately imaged countenances would command the hard positives and negatives). To evade this: Generate triplets on the web. That is select x p and x n (argmax and argmin) from a scaled-down clump (not from the whole dataset) for x a.

Group subtleties: They test preparing information with the end goal that around 40 pictures are chosen for every character for every short clump (to guarantee a significant portrayal of the grapple positive separations), and arbitrarily test negative appearances for every less bunch. Rather than picking the "hardest" positive for a given grapple, they utilized all the stay positive matches inside the group while as yet choosing hard negatives (one to relate to each grapple); they do this since they discovered this prompts an increasingly steady and quicker merging arrangement.

## **5.3 DEEP CONVOLUTIONAL NETWORKS**

In the entirety of our examinations, we teach the CNN making use of Stochastic Gradient Descent (SGD) with popular backdrop and AdaGrad. In many trials, we begin with gaining knowledge of a pace of 0.05 which we lower to settle the model. The fashions are instated from arbitrary, like, and organized on a CPU bunch for 1,000 to 2,000 hours. The decline within the misfortune (and increment in precision) eases back off considerably after 500h of getting ready, yet extra getting ready can at present essentially improve execution. The part α is about to 0.2. We utilized two kinds of designs and look at their exchange offs in more detail within the exploratory area. Their pragmatic contrasts lie inside the distinction of parameters and FLOPS. The satisfactory model is probably distinctive depending upon the application. For example, a model going for walks in the middle of a record can have numerous parameters and require a sizable wide variety of FLOPS, whilst a model jogging on a cell phone desires to have barely any parameters, so it could suit into memory. Every one in all our models utilizes amended direct gadgets as the non-instantly enactment work.

The predominant elegance seemed in Table, includes 1\*1 d convolutional layers, as endorsed in, between the usual convolutional layers of the Zeiler&Fergus design and outcomes in a version 22 layers profound. It has an aggregate of one hundred forty million parameters and requires round 1.6 billion FLOPS for every picture. The 2nd elegance we use depends on Google Net fashion Inception fashions. These models have 20× fewer parameters (around 6.6M-7.5M) and up to 5× fewer FLOPS

Table 5.2.zeiler&Fergus model NN1

Table 1. NN1. This table indicates the shape of our Zeiler&Fergus primarily based model with 1\*1 convolutions propelled by using. The statistics and yield sizes are depicted in columns × cols × #filters. The piece is determined as columns × cols, walk, and the maximum out pooling size as p = 2.(between 500M-1.6B). A portion of those models is drastically faded in size (both profundity and number of channels), with the purpose that they may be run on a cell phone. One, NNS1, has 26M parameters and simply requires 220M FLOPS consistent with the picture. The other, NNS2, has 4.3M parameters and 20M FLOPS. Table2describes NN2 our biggest machine in detail. NN3 is indistinguishable in engineering however has a diminished data length of 160x160. NN4 has a statistics size of just 96x96, in this way diminishing the CPU necessities (285M FLOPS as opposed to 1.6B for NN2). Notwithstanding the length of the diminished record, it doesn't make use of 5x5 convolutions in the better layers because the responsive area is as of now excessively little with the aid of at that point. For the maximum part, we found that the 5x5 convolutions can be evacuated all through with only a minor drop inexactness. The parent looks in any respect for our fashions.

**5.4 DATASETS AND EVALUATION**

We assess our technique on four datasets and apart from tagged Faces within the Wild and YouTube Faces we tend to assess our strategy on the face check task. let's say, given a few of two face footage a square L2 separation limit, D(xi, xj) is used to make a decision the grouping of the same and distinctive. All faces sets (I, j) of a similar character are signified with Psame, though all sets of various personalities are indicated with Pdiff.

Categorizing the arrangement of the obvious point as

TA(d) = {(i, j) ∈ Psame, with D(xi, xj) ≤ d} . (4)

These are that face sets (I, j) those were accurately named the same at edge d. Correspondingly

“FA(d) = {(i, j) ∈ Pdiff, with D(xi, xj) ≤ d}” (5) it is the arrangement of all combines that was mistakenly substituted same

(bogus acknowledge).

The approval rate VAL(d) and the bogus acknowledge rate

FAR(d) for a given face separation d are then characterized as

VAL(d) = |TA(d)| , FAR(d) = |FA(d)| . (6)

It comprises of three individual photograph assortments with an aggregate of around 12k pictures. We figure the FAR and VAL rate overall 12k squared sets of pictures.

Table 5.3.NN2

Subtleties of the NN2 origination manifestation. This model is much indistinguishable from the one delineated in. the 2 important contrasts are the use of L2 pooling instead of max-pooling (m), wherever determined. The pooling is consistently three (besides the last traditional pooling) and equivalent to the convolutional modules within each origination module. If there's a spatial property decrease when the pooling it's indicated with p. 1×1, 3×3, and 5×5 pooling are then coupled to obtaining the last yield.

**5.5. DATASETS FOR ACADEMIC PURPOSE**

Named Faces within the Wild (LFW) is that the accepted scholastic check set for face confirmation. we tend to observe the quality convention for unhampered, marked outside data, and report the mean arrangement exactitude even as the quality mistake of the mean.   
YouTube Faces dB is another dataset that has picked up prominence within the face recognition network. The arrangement is like LFW, however, as hostile checking sets of images, sets of recordings are used.

**CHAPTER 6**

**RESULTS AND DISCUSSIONS**

We use between 100M-200M for preparing face thumbnails which comprise about 8M characters, which is not referenced earlier. A FACE-INDICATOR will run on each of the images and a closed bounding box around each face will be produced. These are resized to the info size of that particular system. Info bytes run with a minimum of 96x96 to a maximum of 224x224 pixels in our trails.

## **6.1. COMPUTATION ACCURACY TRADE-OFF**

Before pitching into the details of more precise experiments Let us discuss the trade-off of accuracy versus no. of FLOPS that the particular model requires.

Figure

Figure 6.1. FLOPS compared with Accuracy trade-off.

It is the trade-off between FLOPS and accuracy for a wide range of dissimilar model dimensions and designs. Here there are four models that we comparatively focus on in our experiments.

|  |  |
| --- | --- |
| architecture | VAL |

|  |  |
| --- | --- |
| NN1 (Zeiler&Fergus 220×220)  NN2 (Inception 224×224)  NN3 (Inception 160×160)  NN4 (Inception 96×96)  NNS1 (mini Inception 165×165)  NNS2 (tiny Inception 140×116) | 87*.*9% ± 1*.*9  89*.*4% ± 1*.*6  88*.*3% ± 1*.*7  82*.*0% ± 2*.*3  82*.*4% ± 2*.*4  51*.*9% ± 2*.*9 |

Table6.1. The architecture of Networks.

The table above thinks about the presentation of our prototypical structures on the holdout test set (see section4.1). Publicized is mean approval rate VAL at the “10E-3” bogus acknowledge rate. Moreover, demonstrated is the standard blunder of the mean over the 5-test parts.

Showing the FLOPS within the graph using x-pivot and the precision at “0.001” bogus acknowledge rate (FAR) on our client named “test-informational index”. This is intriguing to look at the solid relationship between the calculation, a model requires and the accuracy it achieves. The figure structures the five models (NN1, NN2, NN3, NNS1, NNS2) that we are going to examine furthermore.

We had additionally investigated the exactness exchange off-concerning the number of design parameters. In any case, the image isn't as clear all things considered. For example, the Initiation based model NN2 accomplishes a similar exhibition to NN1, however, it just has a twentieth of the constraints. The quantity of FLOPS is equivalent, however. Sooner or later the presentation is relied upon to diminish if the quantity of parameters is decreased additionally. Other model designs may permit an additional decrease without loss of accuracy, much the same as Inception [16] did for this situation.

Figure 6.2. Architecture of Network.

This plot shows the total ROC for the four unique models on our photos test set from section4.2. The sharp drop at “10E-4 FAR” can be clarified by clamor in the ground truth names. The models arranged by execution are: “NN2: 224\*224 information Inception based model”; “NN1: Zeiler&Fergus based system with 1\*1 convolutions”; “NNS1: little Inception-style model with just 220M FLOPS”; “NNS2: minor Inception model with just 20M FLOPS.”

### **6.2 UPSHOT OF CNN MODEL**

We currently talk about the presentation of our 4 chose models in more aspects. From a single viewpoint we take our conventional “Zeiler&Fergus” put together a design with the 1\*1 convolutions to the next hand, we have Inception based models that significantly reduce the model magnitude. Generally speaking, with-in the last execution, the highest models of the two structures accomplish similarly. Be that as it may, a portion of our Inception based models, for sample, NN3, still does a great job executing while primarily decreasing both the F.L.O.P.S. and the model dimensions.

The definite assessment of our own photographs test set has appeared in Figure. Although the biggest model accomplishes an emotional development in precision contrasted with the small NNS2, the last can run with 30ms/picture on a cell phone is as yet exact enough to be utilized in the face bunching. The drastically dropping in the ROC for FAR < 10−4 shows boisterous names in the test information ground truth. At very low bogus acknowledge rates a solitary mislabelled picture could significantly affect the bend.

### **6.3 AFFECTABILITY TO PICTURE QUALITY**

Table 4 depicts the strength of our design over a wide scope of picture sizes. The system is astonishingly powerful regarding JPEG weight and achieves to a JPEG nature of 20. The exhibition drop is little for face thumbnails down to a size of “120x120 pixels” and even at “80x80 pixels”, it shows satisfactory execution. This is imminent because the prepared system was on “220x220 information pictures”. Making this for the lesser goal of appearances increases the range furthermore.

|  |  |
| --- | --- |
| Jpeg pic | aval-rate |

|  |  |
| --- | --- |
| 10 | 67.3% |
| 20 | 81.4% |
| 30 | 83.9% |
| 50 | 85.5% |
| 70 | 86.1% |
| 90 | 86.5% |

|  |  |
| --- | --- |
| #pixels | aval-rate |

|  |  |
| --- | --- |
| 1,600 | 37.8% |
| 6,400 | 79.5% |
| 14,400 | 84.5% |
| 25,600 | 85.7% |
| 65,536 | 86.4% |

Table 6.2. Quality of Image.

The left table gives the effect on the approval rate at “10E-3” accuracy with differing quality of JPEG format. The right table shows how the picture size in pixels affects the approval rate at “10E-3 accuracy”. This complete test was finished with the NN1 on the principal fragments of our test hold-out dataset.

|  |  |
| --- | --- |
| #dimensions | VAL rate |

|  |  |
| --- | --- |
| 64  128  256  512 | 86*.*8% ± 1*.*7  87*.*9% ± 1*.*9  87*.*7% ± 1*.*9  85*.*6% ± 2*.*0 |

Table 6.3. Embedding Dimensionality.

The table above compares the effect of the dimensions to the val rate. In addition to the VAL at “10E-3”, we are also showing the standard error computed across five splits.

### **6.4 EMBEDDING DIMENSIONALITY**

We explored different inserting dimensionalities and chose 128 for all trials other than the correlation reported in Table5. You can expect insertions to perform in any event on a par with the littler ones, nonetheless, it is believable that they require additional training to obtain a similar exactness. Considering all these things, the differences in the exhibition reported in table 5 are irrelevant.

It is noticed, that during the training period a 128-dimensional buoy vector is used which managed the quantizing to 128-bytes without losing the exactness. Each face is tested by the same 128-dimensional vector which is perfect for scoping of large groups of faces.

**6.4.1 EXTENT OF TRAINING DATA**

Table6shows the effect of using a lot of training data. Due to time imperatives, this valuation was run on a littler model; the impact might be much bigger on bigger models. Utilizing a huge number of models brings about an away from of exactness on our photograph trial set from section4.2. Compared with just a huge number of pictures the reduction in mistake is 60%. Utilizing other pictures may give a little lift but the improvement decreases.

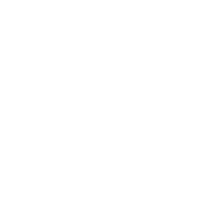
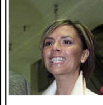
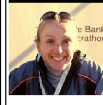
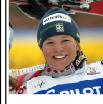
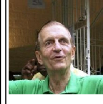
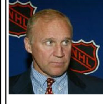
|  |  |
| --- | --- |
| #training\_images | V.A.L |

|  |  |
| --- | --- |
| 2,600,000 | 76.3% |

|  |  |
| --- | --- |
| 26,000,000 | 85.1% |
| 52,000,000 | 85.1% |
| 60,000,000 | 86.2% |

Table 6.4. Data Size of Training data.

The above table compares the performance for the 700h of training for a small model which of 96x96 pixel input. The design architecture is similar to that of NN2 except for the 5x5 convolutions we see in the Inception modules.



**Please accept**

**Please reject**

### 

### **6.4.2 PERFORMANCE ON LFW**

We are going to assess our model on LFW utilizing the standard convention for unlimited, named outside data. Nine training parts are utilized to choose the L2-separation limit. The order is then performed on the 10th test split. The chose ideal limit is just about“1.242” for all test parts aside from split 8th(1.256).

We assessed our model in these 2 ways:

1. The immovable focus yield of the LFW gave thumbnail.

2. An exclusive face indicator will run on the given LFW thumbnails. On the off chance that it neglects to adjust the face (this occurs for two pictures), the LFW arrangement is used.

The figure gives a review of all the disappointment cases. The bogus acknowledges on the top just as bogus rejects at the base. We appreciate an order precision of 98.87% 0.15 when utilizing the immovable focus crop depicted in and the record-breaking “99.63% 0.09” standard mistake of the mean when utilizing the additional face arrangement. This lessens blunder revealed for Deep Face in by greater than a factor of 7 and the past best in class announced for DeepId2+ in [by 30%.

**6.4.3 PERFORMANCE ON YOUTUBE FACES DATABASE**

This utilizes the normal closeness of all sets of the first hundred edges that our face identifier recognizes in every video. A grouping precision of “95.12% 0.39” is obtained. Utilizing the first thousand casings brings about 95.18%. Contrasted with “91.4%” who moreover assess 100 casings for every video we can reduce the mistake rate by 50%. DeepId2+ accomplished “93.2%” and this strategy we used reduces it by 30percent. Which is better when compared.

**6.4.4 FACE CLUSTERING**

Our reduced implanting fits be utilized to bunch a client's photographs into gatherings of individuals with a similar personality. The imperatives in task forced by clustering faces, contrasted with the unadulterated confirmation task, lead to really astonishing outcomes. Figure7shows one bunch in a client's photograph assortment produced utilizing agglomerative clustering. It is away from the unbelievable invariance to impediment, lighting, present, and even age.



## Figure 6.4 face clustering

**CHAPTER-7**

## **CONCLUSIONS AND FUTURESCOPE**

## 

**7.1 CONCLUSION:**

We proposed to assemble a superior, adaptable, nimble, and ease facial recognition framework. We differentiated the proposed approach into a few smaller sub-ventures. In the first place, we considered the neural system and convolutional neural system. In the way of obtaining a profound learning method, we used the Siamese system which will prepare the neural networks accordingly. At that point we look at and analyze the accessible open-source informational collection, we picked the ORL dataset and prepared the model using GPU. This model will take a human picture and convert it to a vector. Similarly many vectors are created and these are contrasted with one another to find they both are of the same person.

At that point we did the investigation, think about, structure and fabricate a framework which is going to work with neural system model. This framework uses customer server design. The installed GPU is utilized on the server-side to give elite performance. We additionally isolated the fundamental parts of the framework to make it adaptable and versatile to make it different. We aslos utilized the non-square and asynchronies highlights of “Node.JS” to expand the framework's simultaneousness. Since the whole framework is modularized, it very well may be utilized in various spaces, subsequently decreased the improvement cost.

**7.2 FUTURE SCOPE:**

During the manufacturing of neural system model, there are numerous constraints which are tuned to obtain a model suitable for execution. We can further tune this for better accuracy.

Moreover, for a prepared base model, we can re-train it utilizing a particular dataset. There is another approach to expand the framework operation by catching the individual faces which can go up to 3000 individuals. Which helps in better utilisation. We can use and retrain this component in the framework.

**REFERENCES**

[1]  Y. Bengio, J. Louradour, R. Collobert, and J. Weston.Curriculum learning. In Proc. of ICML, New York, NY, USA, 2009. 2

[2] D.Chen, X.Cao, L.Wang, F.Wen, and J. Sun. Bayesian face revisited: A joint formulation. In Proc. ECCV, 2012. 2

[3]  D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In Proc. ECCV, 2014. 7

[4]  J. Dean, G. Corrado, R. Monga, K.Chen, M.Devin, M.Mao, M.Ranzato, A.Senior,P.Tuck-er, K.Yang, Q.V.Le, and A.Y.Ng. Large scale distributed deep networks.In P.Bartlett, F.Pereira, C.Burges,L.Bottou, and K. Weinberger, editors, NIPS, pages 1232–1240. 2012. 10

[5] J. Duchi,E.Hazan,and Y.Singer. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 12:2121–2159, July 2011. 4

[6]  I. J. Goodfellow,D.Warde farley, M. Mirza, A. Courville, and Y. Bengio. Maxout netwo-rks. In In ICML, 2013. 4

[7]  G.B.Huang, M.Ramesh, T.Berg, and E. Learned Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environ-ments. Technical Report 0749,University of Massachusetts, Amherst, October 2007. 5

[8] Y. LeCun, B.Boser, J. S. Denker, D. Henderson, R. E.Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation,

1(4):541–551, Dec. 1989. 2, 4

[9]  M. Lin, Q. Chen, and S. Yan. Network in network. CoRR, abs/1312.4400, 2013. 2, 4, 6

[10] C. Lu and X. Tang. Surpassing human-level face verification performance on LFW with gaussianface. CoRR, abs/1404.3840, 2014. 1

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. Nature, 1986. 2,4

[12] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In S. Thrun, L. Saul, and B. Schölkopf, editors, NIPS, pages 41–48. MIT Press, 2004. 2

[13] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In In Proc. FG, 2002. 2

[14] Y. Sun, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. CoRR, abs/1406.4773, 2014. 1, 2, 3

[15] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. CoRR, abs/1412.1265, 2014. 1, 2, 5

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014. 2, 3, 4, 5, 6, 10

[17] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In IEEE Conf. on CVPR, 2014. 1, 2, 5, 7, 8, 9

**Appendix:**

# coding: utf-8

# # Face Recognition

#

# In this assignment, you will build a face recognition system. Many of the ideas presented here are from [FaceNet](https://arxiv.org/pdf/1503.03832.pdf). In lecture, we also talked about [DeepFace](https://research.fb.com/wp-content/uploads/2016/11/deepface-closing-the-gap-to-human-level-performance-in-face-verification.pdf).

#

# Face recognition problems commonly fall into two categories:

#

# - \*\*Face Verification\*\* - "is this the claimed person?". For example, at some airports, you can pass through customs by letting a system scan your passport and then verifying that you (the person carrying the passport) are the correct person. A mobile phone that unlocks using your face is also using face verification. This is a 1:1 matching problem.

# - \*\*Face Recognition\*\* - "who is this person?". For example, the video lecture showed a [face recognition video](https://www.youtube.com/watch?v=wr4rx0Spihs) of Baidu employees entering the office without needing to otherwise identify themselves. This is a 1:K matching problem.

#

# FaceNet learns a neural network that encodes a face image into a vector of 128 numbers. By comparing two such vectors, you can then determine if two pictures are of the same person.

#

# \*\*In this assignment, you will:\*\*

# - Implement the triplet loss function

# - Use a pretrained model to map face images into 128-dimensional encodings

# - Use these encodings to perform face verification and face recognition

#

# #### Channels-first notation

#

# \* In this exercise, we will be using a pre-trained model which represents ConvNet activations using a \*\*"channels first"\*\* convention, as opposed to the "channels last" convention used in lecture and previous programming assignments.

# \* In other words, a batch of images will be of shape $(m, n\_C, n\_H, n\_W)$ instead of $(m, n\_H, n\_W, n\_C)$.

# \* Both of these conventions have a reasonable amount of traction among open-source implementations; there isn't a uniform standard yet within the deep learning community.

# ## <font color='darkblue'>Updates</font>

#

# #### If you were working on the notebook before this update...

# \* The current notebook is version "3a".

# \* You can find your original work saved in the notebook with the previous version name ("v3")

# \* To view the file directory, go to the menu "File->Open", and this will open a new tab that shows the file directory.

#

# #### List of updates

# \* `triplet\_loss`: Additional Hints added.

# \* `verify`: Hints added.

# \* `who\_is\_it`: corrected hints given in the comments.

# \* Spelling and formatting updates for easier reading.

#

# #### Load packages

# Let's load the required packages.

# In[ ]:

from keras.models import Sequential

from keras.layers import Conv2D, ZeroPadding2D, Activation, Input, concatenate

from keras.models import Model

from keras.layers.normalization import BatchNormalization

from keras.layers.pooling import MaxPooling2D, AveragePooling2D

from keras.layers.merge import Concatenate

from keras.layers.core import Lambda, Flatten, Dense

from keras.initializers import glorot\_uniform

from keras.engine.topology import Layer

from keras import backend as K

K.set\_image\_data\_format('channels\_first')

import cv2

import os

import numpy as np

from numpy import genfromtxt

import pandas as pd

import tensorflow as tf

from fr\_utils import \*

from inception\_blocks\_v2 import \*

get\_ipython().magic('matplotlib inline')

get\_ipython().magic('load\_ext autoreload')

get\_ipython().magic('autoreload 2')

np.set\_printoptions(threshold=np.nan)

# ## 0 - Naive Face Verification

#

# In Face Verification, you're given two images and you have to determine if they are of the same person. The simplest way to do this is to compare the two images pixel-by-pixel. If the distance between the raw images are less than a chosen threshold, it may be the same person!

#

# <img src="images/pixel\_comparison.png" style="width:380px;height:150px;">

# <caption><center> <u> <font color='purple'> \*\*Figure 1\*\* </u></center></caption>

# \* Of course, this algorithm performs really poorly, since the pixel values change dramatically due to variations in lighting, orientation of the person's face, even minor changes in head position, and so on.

# \* You'll see that rather than using the raw image, you can learn an encoding, $f(img)$.

# \* By using an encoding for each image, an element-wise comparison produces a more accurate judgement as to whether two pictures are of the same person.

# ## 1 - Encoding face images into a 128-dimensional vector

#

# ### 1.1 - Using a ConvNet to compute encodings

#

# The FaceNet model takes a lot of data and a long time to train. So following common practice in applied deep learning, let's load weights that someone else has already trained. The network architecture follows the Inception model from [Szegedy \*et al.\*](https://arxiv.org/abs/1409.4842). We have provided an inception network implementation. You can look in the file `inception\_blocks\_v2.py` to see how it is implemented (do so by going to "File->Open..." at the top of the Jupyter notebook. This opens the file directory that contains the '.py' file).

# The key things you need to know are:

#

# - This network uses 96x96 dimensional RGB images as its input. Specifically, inputs a face image (or batch of $m$ face images) as a tensor of shape $(m, n\_C, n\_H, n\_W) = (m, 3, 96, 96)$

# - It outputs a matrix of shape $(m, 128)$ that encodes each input face image into a 128-dimensional vector

#

# Run the cell below to create the model for face images.

# In[ ]:

FRmodel = faceRecoModel(input\_shape=(3, 96, 96))

# In[ ]:

print("Total Params:", FRmodel.count\_params())

# \*\* Expected Output \*\*

# <table>

# <center>

# Total Params: 3743280

# </center>

# </table>

#

# By using a 128-neuron fully connected layer as its last layer, the model ensures that the output is an encoding vector of size 128. You then use the encodings to compare two face images as follows:

#

# <img src="images/distance\_kiank.png" style="width:680px;height:250px;">

# <caption><center> <u> <font color='purple'> \*\*Figure 2\*\*: <br> </u> <font color='purple'> By computing the distance between two encodings and thresholding, you can determine if the two pictures represent the same person</center></caption>

#

# So, an encoding is a good one if:

# - The encodings of two images of the same person are quite similar to each other.

# - The encodings of two images of different persons are very different.

#

# The triplet loss function formalizes this, and tries to "push" the encodings of two images of the same person (Anchor and Positive) closer together, while "pulling" the encodings of two images of different persons (Anchor, Negative) further apart.

#

# <img src="images/triplet\_comparison.png" style="width:280px;height:150px;">

# <br>

# <caption><center> <u> <font color='purple'> \*\*Figure 3\*\*: <br> </u> <font color='purple'> In the next part, we will call the pictures from left to right: Anchor (A), Positive (P), Negative (N) </center></caption>

#

#

# ### 1.2 - The Triplet Loss

#

# For an image $x$, we denote its encoding $f(x)$, where $f$ is the function computed by the neural network.

#

# <img src="images/f\_x.png" style="width:380px;height:150px;">

#

# <!--

# We will also add a normalization step at the end of our model so that $\mid \mid f(x) \mid┬á\mid\_2 = 1$ (means the vector of encoding should be of norm 1).

# !-->

#

# Training will use triplets of images $(A, P, N)$:

#

# - A is an "Anchor" image--a picture of a person.

# - P is a "Positive" image--a picture of the same person as the Anchor image.

# - N is a "Negative" image--a picture of a different person than the Anchor image.

#

# These triplets are picked from our training dataset. We will write $(A^{(i)}, P^{(i)}, N^{(i)})$ to denote the $i$-th training example.

#

# You'd like to make sure that an image $A^{(i)}$ of an individual is closer to the Positive $P^{(i)}$ than to the Negative image $N^{(i)}$) by at least a margin $\alpha$:

#

# $$\mid \mid f(A^{(i)}) - f(P^{(i)}) \mid \mid\_2^2 + \alpha < \mid \mid f(A^{(i)}) - f(N^{(i)}) \mid \mid\_2^2$$

#

# You would thus like to minimize the following "triplet cost":

#

# $$\mathcal{J} = \sum^{m}\_{i=1} \large[ \small \underbrace{\mid \mid f(A^{(i)}) - f(P^{(i)}) \mid \mid\_2^2}\_\text{(1)} - \underbrace{\mid \mid f(A^{(i)}) - f(N^{(i)}) \mid \mid\_2^2}\_\text{(2)} + \alpha \large ] \small\_+┬á\tag{3}$$

#

# Here, we are using the notation "$[z]\_+$" to denote $max(z,0)$.

#

# Notes:

# - The term (1) is the squared distance between the anchor "A" and the positive "P" for a given triplet; you want this to be small.

# - The term (2) is the squared distance between the anchor "A" and the negative "N" for a given triplet, you want this to be relatively large. It has a minus sign preceding it because minimizing the negative of the term is the same as maximizing that term.

# - $\alpha$ is called the margin. It is a hyperparameter that you pick manually. We will use $\alpha = 0.2$.

#

# Most implementations also rescale the encoding vectors to haven L2 norm equal to one (i.e., $\mid \mid f(img)\mid \mid\_2$=1); you won't have to worry about that in this assignment.

#

# \*\*Exercise\*\*: Implement the triplet loss as defined by formula (3). Here are the 4 steps:

# 1. Compute the distance between the encodings of "anchor" and "positive": $\mid \mid f(A^{(i)}) - f(P^{(i)}) \mid \mid\_2^2$

# 2. Compute the distance between the encodings of "anchor" and "negative": $\mid \mid f(A^{(i)}) - f(N^{(i)}) \mid \mid\_2^2$

# 3. Compute the formula per training example: $ \mid \mid f(A^{(i)}) - f(P^{(i)}) \mid \mid\_2^2 - \mid \mid f(A^{(i)}) - f(N^{(i)}) \mid \mid\_2^2 + \alpha$

# 3. Compute the full formula by taking the max with zero and summing over the training examples:

# $$\mathcal{J} = \sum^{m}\_{i=1} \large[ \small \mid \mid f(A^{(i)}) - f(P^{(i)}) \mid \mid\_2^2 - \mid \mid f(A^{(i)}) - f(N^{(i)}) \mid \mid\_2^2+ \alpha \large ] \small\_+┬á\tag{3}$$

#

# #### Hints

# \* Useful functions: `tf.reduce\_sum()`, `tf.square()`, `tf.subtract()`, `tf.add()`, `tf.maximum()`.

# \* For steps 1 and 2, you will sum over the entries of $\mid \mid f(A^{(i)}) - f(P^{(i)}) \mid \mid\_2^2$ and $\mid \mid f(A^{(i)}) - f(N^{(i)}) \mid \mid\_2^2$.

# \* For step 4 you will sum over the training examples.

#

# #### Additional Hints

# \* Recall that the square of the L2 norm is the sum of the squared differences: $||x - y||\_{2}^{2} = \sum\_{i=1}^{N}(x\_{i} - y\_{i})^{2}$

# \* Note that the `anchor`, `positive` and `negative` encodings are of shape `(m,128)`, where m is the number of training examples and 128 is the number of elements used to encode a single example.

# \* For steps 1 and 2, you will maintain the number of `m` training examples and sum along the 128 values of each encoding.

# [tf.reduce\_sum](https://www.tensorflow.org/api\_docs/python/tf/math/reduce\_sum) has an `axis` parameter. This chooses along which axis the sums are applied.

# \* Note that one way to choose the last axis in a tensor is to use negative indexing (`axis=-1`).

# \* In step 4, when summing over training examples, the result will be a single scalar value.

# \* For `tf.reduce\_sum` to sum across all axes, keep the default value `axis=None`.

# In[ ]:

# GRADED FUNCTION: triplet\_loss

def triplet\_loss(y\_true, y\_pred, alpha = 0.2):

"""

Implementation of the triplet loss as defined by formula (3)

Arguments:

y\_true -- true labels, required when you define a loss in Keras, you don't need it in this function.

y\_pred -- python list containing three objects:

anchor -- the encodings for the anchor images, of shape (None, 128)

positive -- the encodings for the positive images, of shape (None, 128)

negative -- the encodings for the negative images, of shape (None, 128)

Returns:

loss -- real number, value of the loss

"""

anchor, positive, negative = y\_pred[0], y\_pred[1], y\_pred[2]

### START CODE HERE ### (Γëê 4 lines)

# Step 1: Compute the (encoding) distance between the anchor and the positive

pos\_dist = None

# Step 2: Compute the (encoding) distance between the anchor and the negative

neg\_dist = None

# Step 3: subtract the two previous distances and add alpha.

basic\_loss = None

# Step 4: Take the maximum of basic\_loss and 0.0. Sum over the training examples.

loss = None

### END CODE HERE ###

return loss

# In[ ]:

with tf.Session() as test:

tf.set\_random\_seed(1)

y\_true = (None, None, None)

y\_pred = (tf.random\_normal([3, 128], mean=6, stddev=0.1, seed = 1),

tf.random\_normal([3, 128], mean=1, stddev=1, seed = 1),

tf.random\_normal([3, 128], mean=3, stddev=4, seed = 1))

loss = triplet\_loss(y\_true, y\_pred)

print("loss = " + str(loss.eval()))

# \*\*Expected Output\*\*:

#

# <table>

# <tr>

# <td>

# \*\*loss\*\*

# </td>

# <td>

# 528.143

# </td>

# </tr>

#

# </table>

# ## 2 - Loading the pre-trained model

#

# FaceNet is trained by minimizing the triplet loss. But since training requires a lot of data and a lot of computation, we won't train it from scratch here. Instead, we load a previously trained model. Load a model using the following cell; this might take a couple of minutes to run.

# In[ ]:

FRmodel.compile(optimizer = 'adam', loss = triplet\_loss, metrics = ['accuracy'])

load\_weights\_from\_FaceNet(FRmodel)

# Here are some examples of distances between the encodings between three individuals:

#

# <img src="images/distance\_matrix.png" style="width:380px;height:200px;">

# <br>

# <caption><center> <u> <font color='purple'> \*\*Figure 4\*\*:</u> <br> <font color='purple'> Example of distance outputs between three individuals' encodings</center></caption>

#

# Let's now use this model to perform face verification and face recognition!

# ## 3 - Applying the model

# You are building a system for an office building where the building manager would like to offer facial recognition to allow the employees to enter the building.

#

# You'd like to build a \*\*Face verification\*\* system that gives access to the list of people who live or work there. To get admitted, each person has to swipe an ID card (identification card) to identify themselves at the entrance. The face recognition system then checks that they are who they claim to be.

# ### 3.1 - Face Verification

#

# Let's build a database containing one encoding vector for each person who is allowed to enter the office. To generate the encoding we use `img\_to\_encoding(image\_path, model)`, which runs the forward propagation of the model on the specified image.

#

# Run the following code to build the database (represented as a python dictionary). This database maps each person's name to a 128-dimensional encoding of their face.

# In[ ]:

database = {}

database["danielle"] = img\_to\_encoding("images/danielle.png", FRmodel)

database["younes"] = img\_to\_encoding("images/younes.jpg", FRmodel)

database["tian"] = img\_to\_encoding("images/tian.jpg", FRmodel)

database["andrew"] = img\_to\_encoding("images/andrew.jpg", FRmodel)

database["kian"] = img\_to\_encoding("images/kian.jpg", FRmodel)

database["dan"] = img\_to\_encoding("images/dan.jpg", FRmodel)

database["sebastiano"] = img\_to\_encoding("images/sebastiano.jpg", FRmodel)

database["bertrand"] = img\_to\_encoding("images/bertrand.jpg", FRmodel)

database["kevin"] = img\_to\_encoding("images/kevin.jpg", FRmodel)

database["felix"] = img\_to\_encoding("images/felix.jpg", FRmodel)

database["benoit"] = img\_to\_encoding("images/benoit.jpg", FRmodel)

database["arnaud"] = img\_to\_encoding("images/arnaud.jpg", FRmodel)

# Now, when someone shows up at your front door and swipes their ID card (thus giving you their name), you can look up their encoding in the database, and use it to check if the person standing at the front door matches the name on the ID.

#

# \*\*Exercise\*\*: Implement the verify() function which checks if the front-door camera picture (`image\_path`) is actually the person called "identity". You will have to go through the following steps:

# 1. Compute the encoding of the image from `image\_path`.

# 2. Compute the distance between this encoding and the encoding of the identity image stored in the database.

# 3. Open the door if the distance is less than 0.7, else do not open it.

#

#

# \* As presented above, you should use the L2 distance [np.linalg.norm](https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.norm.html).

# \* (Note: In this implementation, compare the L2 distance, not the square of the L2 distance, to the threshold 0.7.)

#

# #### Hints

# \* `identity` is a string that is also a key in the `database` dictionary.

# \* `img\_to\_encoding` has two parameters: the `image\_path` and `model`.

# In[ ]:

# GRADED FUNCTION: verify

def verify(image\_path, identity, database, model):

"""

Function that verifies if the person on the "image\_path" image is "identity".

Arguments:

image\_path -- path to an image

identity -- string, name of the person you'd like to verify the identity. Has to be an employee who works in the office.

database -- python dictionary mapping names of allowed people's names (strings) to their encodings (vectors).

model -- your Inception model instance in Keras

Returns:

dist -- distance between the image\_path and the image of "identity" in the database.

door\_open -- True, if the door should open. False otherwise.

"""

### START CODE HERE ###

# Step 1: Compute the encoding for the image. Use img\_to\_encoding() see example above. (Γëê 1 line)

encoding = None

# Step 2: Compute distance with identity's image (Γëê 1 line)

dist = None

# Step 3: Open the door if dist < 0.7, else don't open (Γëê 3 lines)

if None:

print("It's " + str(identity) + ", welcome in!")

door\_open = None

else:

print("It's not " + str(identity) + ", please go away")

door\_open = None

### END CODE HERE ###

return dist, door\_open

# Younes is trying to enter the office and the camera takes a picture of him ("images/camera\_0.jpg"). Let's run your verification algorithm on this picture:

#

# <img src="images/camera\_0.jpg" style="width:100px;height:100px;">

# In[ ]:

verify("images/camera\_0.jpg", "younes", database, FRmodel)

# \*\*Expected Output\*\*:

#

# <table>

# <tr>

# <td>

# \*\*It's younes, welcome in!\*\*

# </td>

# <td>

# (0.65939283, True)

# </td>

# </tr>

#

# </table>

# Benoit, who does not work in the office, stole Kian's ID card and tried to enter the office. The camera took a picture of Benoit ("images/camera\_2.jpg). Let's run the verification algorithm to check if benoit can enter.

# <img src="images/camera\_2.jpg" style="width:100px;height:100px;">

# In[ ]:

verify("images/camera\_2.jpg", "kian", database, FRmodel)

# \*\*Expected Output\*\*:

#

# <table>

# <tr>

# <td>

# \*\*It's not kian, please go away\*\*

# </td>

# <td>

# (0.86224014, False)

# </td>

# </tr>

#

# </table>

# ### 3.2 - Face Recognition

#

# Your face verification system is mostly working well. But since Kian got his ID card stolen, when he came back to the office the next day and couldn't get in!

#

# To solve this, you'd like to change your face verification system to a face recognition system. This way, no one has to carry an ID card anymore. An authorized person can just walk up to the building, and the door will unlock for them!

#

# You'll implement a face recognition system that takes as input an image, and figures out if it is one of the authorized persons (and if so, who). Unlike the previous face verification system, we will no longer get a person's name as one of the inputs.

#

# \*\*Exercise\*\*: Implement `who\_is\_it()`. You will have to go through the following steps:

# 1. Compute the target encoding of the image from image\_path

# 2. Find the encoding from the database that has smallest distance with the target encoding.

# - Initialize the `min\_dist` variable to a large enough number (100). It will help you keep track of what is the closest encoding to the input's encoding.

# - Loop over the database dictionary's names and encodings. To loop use `for (name, db\_enc) in database.items()`.

# - Compute the L2 distance between the target "encoding" and the current "encoding" from the database.

# - If this distance is less than the min\_dist, then set `min\_dist` to `dist`, and `identity` to `name`.

# In[ ]:

# GRADED FUNCTION: who\_is\_it

def who\_is\_it(image\_path, database, model):

"""

Implements face recognition for the office by finding who is the person on the image\_path image.

Arguments:

image\_path -- path to an image

database -- database containing image encodings along with the name of the person on the image

model -- your Inception model instance in Keras

Returns:

min\_dist -- the minimum distance between image\_path encoding and the encodings from the database

identity -- string, the name prediction for the person on image\_path

"""

### START CODE HERE ###

## Step 1: Compute the target "encoding" for the image. Use img\_to\_encoding() see example above. ## (Γëê 1 line)

encoding = None

## Step 2: Find the closest encoding ##

# Initialize "min\_dist" to a large value, say 100 (Γëê1 line)

min\_dist = None

# Loop over the database dictionary's names and encodings.

for (name, db\_enc) in None:

# Compute L2 distance between the target "encoding" and the current db\_enc from the database. (Γëê 1 line)

dist = None

# If this distance is less than the min\_dist, then set min\_dist to dist, and identity to name. (Γëê 3 lines)

if None:

min\_dist = None

identity = None

### END CODE HERE ###

if min\_dist > 0.7:

print("Not in the database.")

else:

print ("it's " + str(identity) + ", the distance is " + str(min\_dist))

return min\_dist, identity

# Younes is at the front-door and the camera takes a picture of him ("images/camera\_0.jpg"). Let's see if your who\_it\_is() algorithm identifies Younes.

# In[ ]:

who\_is\_it("images/camera\_0.jpg", database, FRmodel)

# \*\*Expected Output\*\*:

#

# <table>

# <tr>

# <td>

# \*\*it's younes, the distance is 0.659393\*\*

# </td>

# <td>

# (0.65939283, 'younes')

# </td>

# </tr>

#

# </table>

# You can change "`camera\_0.jpg`" (picture of younes) to "`camera\_1.jpg`" (picture of bertrand) and see the result.

# #### Congratulations!

#

# \* Your face recognition system is working well! It only lets in authorized persons, and people don't need to carry an ID card around anymore!

# \* You've now seen how a state-of-the-art face recognition system works.

#

# #### Ways to improve your facial recognition model

# Although we won't implement it here, here are some ways to further improve the algorithm:

# - Put more images of each person (under different lighting conditions, taken on different days, etc.) into the database. Then given a new image, compare the new face to multiple pictures of the person. This would increase accuracy.

# - Crop the images to just contain the face, and less of the "border" region around the face. This preprocessing removes some of the irrelevant pixels around the face, and also makes the algorithm more robust.

#

# ## Key points to remember

# - Face verification solves an easier 1:1 matching problem; face recognition addresses a harder 1:K matching problem.

# - The triplet loss is an effective loss function for training a neural network to learn an encoding of a face image.

# - The same encoding can be used for verification and recognition. Measuring distances between two images' encodings allows you to determine whether they are pictures of the same person.

# Congrats on finishing this assignment!

#

# ### References:

# - Florian Schroff, Dmitry Kalenichenko, James Philbin (2015). [FaceNet: A Unified Embedding for Face Recognition and Clustering](https://arxiv.org/pdf/1503.03832.pdf)

# - Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf (2014). [DeepFace: Closing the gap to human-level performance in face verification](https://research.fb.com/wp-content/uploads/2016/11/deepface-closing-the-gap-to-human-level-performance-in-face-verification.pdf)

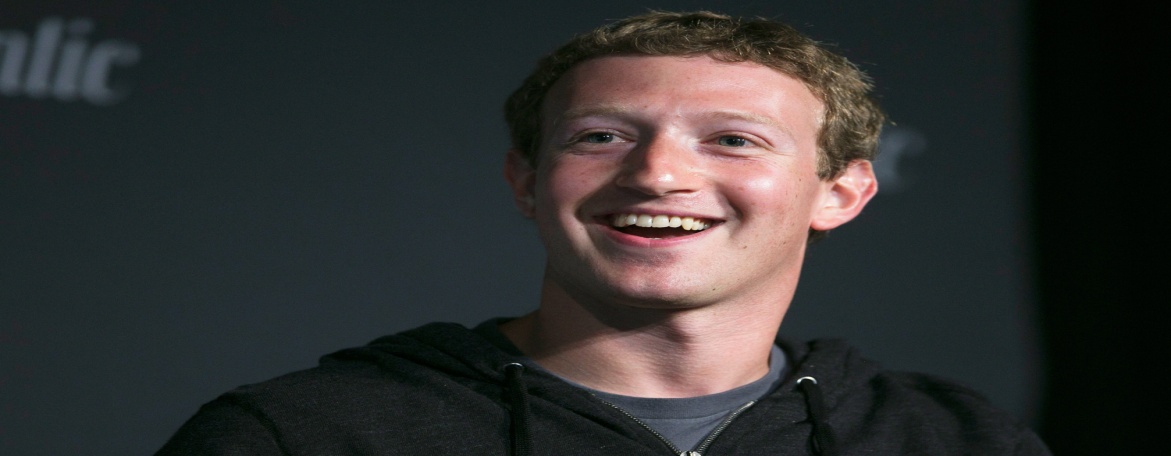
# - The pretrained model we use is inspired by Victor Sy Wang's implementation and was loaded using his code: https://github.com/iwantooxxoox/Keras-OpenFace.

# - Our implementation also took a lot of inspiration from the official FaceNet github repository: https://github.com/davidsandberg/facenet

**OUTPUTS:**

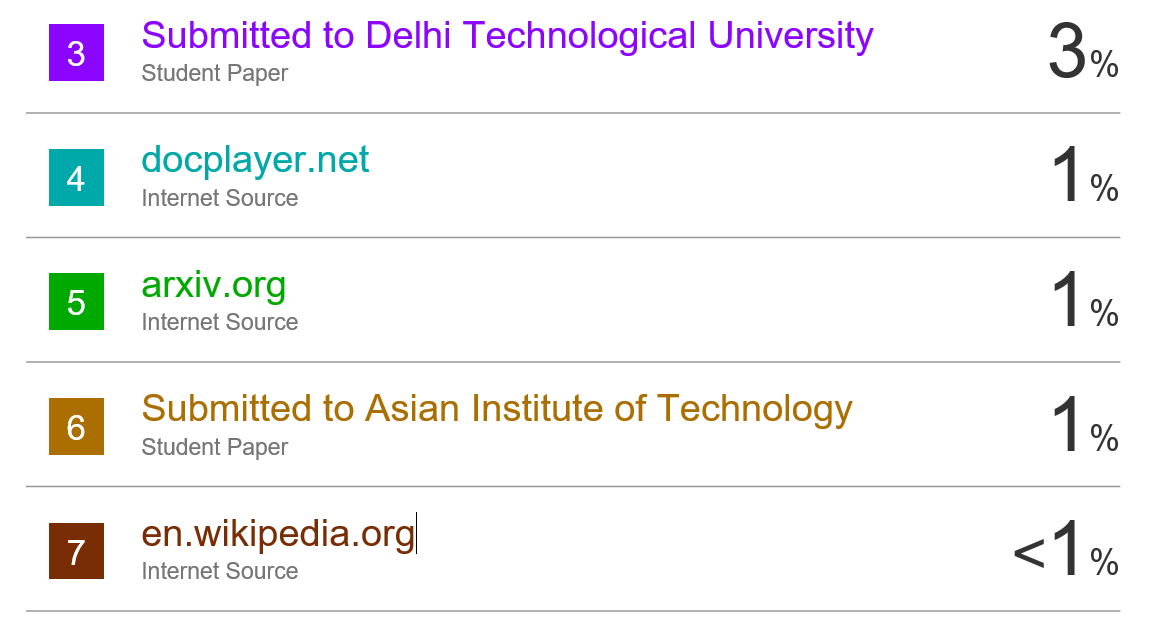
 

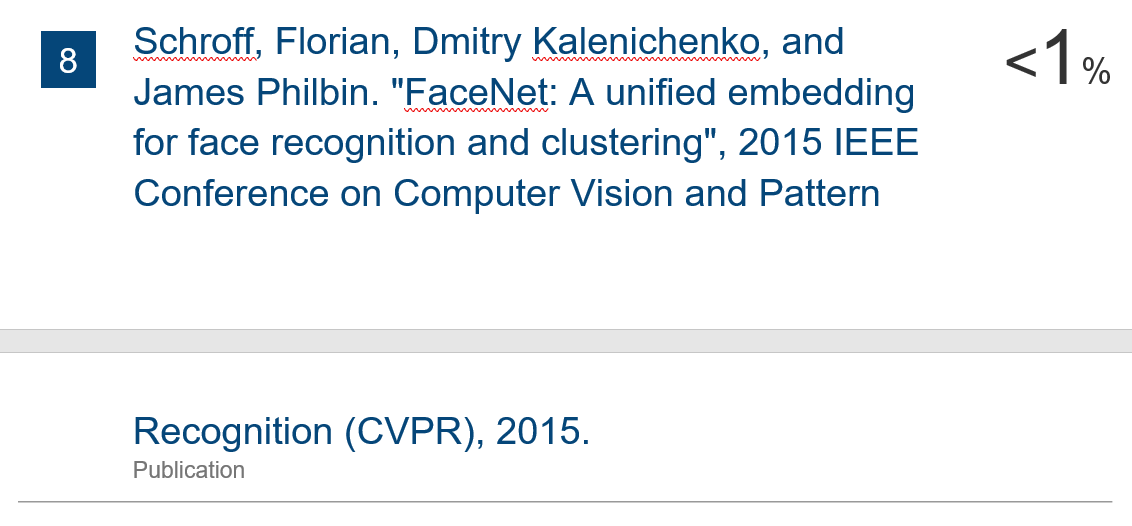
Bill gates Larry page



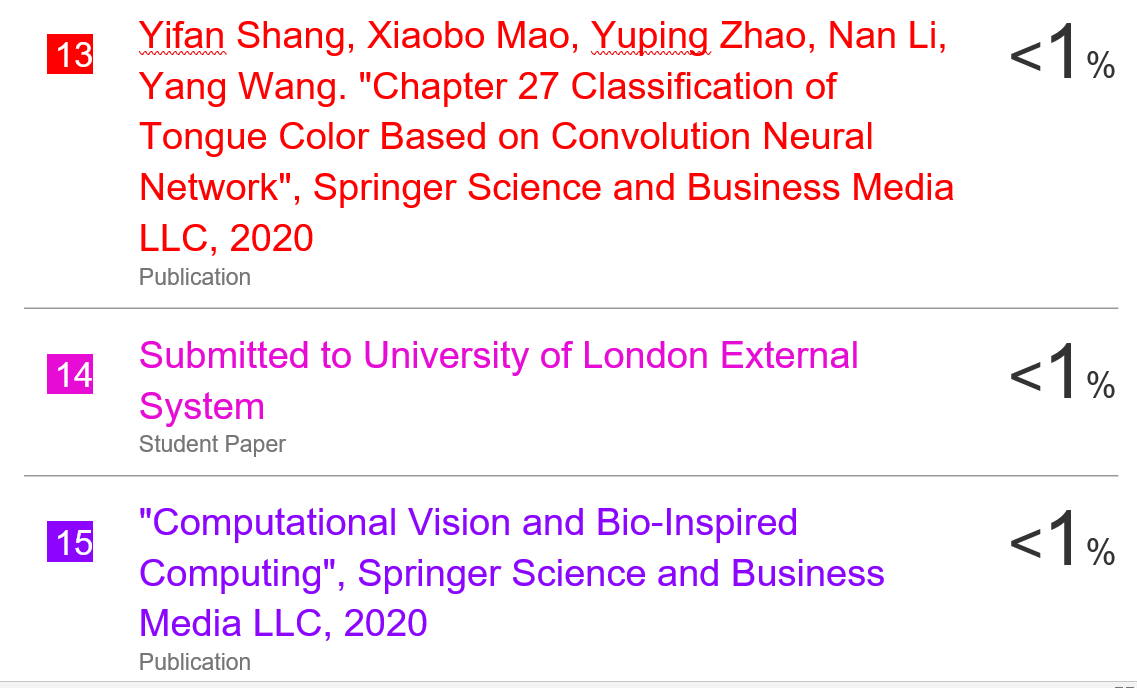
Mark Zuckerberg

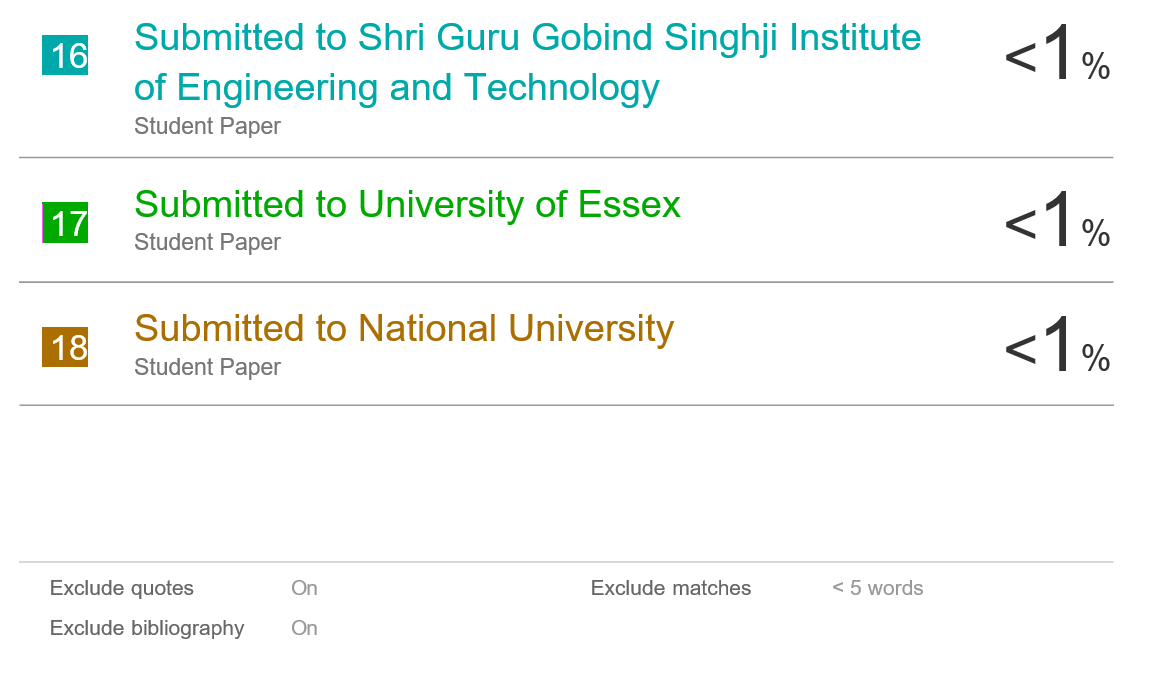












**CONFERENCE PAPER:**

**VGG-NETWORK BASED DEEP CONVOLUTED FACIAL RECOGNITION**

Shruti Bhargava Choubey1, K. Praveen kumar2, K. Vinay kumar3, MD. Abu Sufian4

1Associate professor, ECE department, Sreenidhi institute of Science and Technology, Yamnampet, Medchal

2B.tech 4th year, ECE department, Sreenidhi institute of Science and Technology, Yamnampet, Medchal

3B.tech 4th year, ECE department, Sreenidhi institute of Science and Technology, Yamnampet, Medchal

4B.tech 4th year, ECE department, Sreenidhi institute of Science and Technology, Yamnampet, Medchal

**ABSTRACT:**

Recently Convolution neural network of Deep Learning provided promising results in the development of facial recognition. However, there are no certain techniques which prove its greater ability to perform these tasks. The question about How to create a good architecture still remains as unanswered. The studies only depict us the results they are producing but no the process going on backend. In this paper we designed a model in which the tasks are easily replicated any number of times. Convolution neural networks work better for facial recognition system. We call it as CNN-FRS. Generally, there are methods which train the model by using private database but here we are using a public database LFW (labeled faces in the wild) contradictory to it. We propose 3 CNN architectures which are trained using these LFW, these are compared against each other and evaluates the effect of each architecture.

**Keywords:** Convolution neural networks, LFW, backend, Facial recognition;

1. **INTRODUCTION:**

Due to increase in usage of biometrics for several applications human face recognition has become a challenging task to the computer vision as the pose, illumination and other factors vary from inbuilt data.

In recent years deep learning became the most useful technique to obtain this method as it takes raw data and convolve them into multiple levels which are used in detecting high-level or low-level data representation from labeled or unlabeled data for determining and distinguishing their underlying patterns. Optimizing these millions of parameters by deep learning require millions of training samples and usage of high computational hardware such as Graphical Processing unit (GPU).

Transfer learning can be obtained in two different approaches. The first approach is fine tuning the pre trained data set with a new set using back propagation. This method is best for large datasets since fine-tuning with fewer samples may lead to over fitting. In second approach, the learned weights are directly extracted to classify features.

In this paper, the higher layer portion of the learned weights is pre-trained over a larger dataset for facial recognition. They are extracted using two deep convolutional neural networks of FGG face and Lightened CNN. We selected the above two methods as they proved to be most successful in facial recognition.

1. **LITERATURE SURVEY:**

The article named “Robust Realtime Object Detection” is very most every now and again refered to article in a progression of article’s by “Viola” which makes face discovery genuinely useful. We can find out around a few face discovery techniques and calculation’s from the above distribution. The article named “Fast revolution in-variant multisee face identification” dependend’s on the genuine adaboost just because genuine adaboost applied to question recognition, and proposed an increasingly develop and functionalamultiface location ,home structured referenced on a course structure upgrades likewise have great outcomes.

More than 3 paper’s have examined about a face location and the face tracking issues. As indicated by the exploration bring about these papers, we can make constant face identification frameworks. The important feature is to detect size and the position of face in the video or image but in regard to tracking it is important to determine the similarity between the faces in the casing.

**21 Facial Markers:** In the 1970s, these scientists Harmon, Goldstein, and Lesk made the manual recognition more efficient by scanning the 21 facial features including lipsticks and hair color.

**Eigen faces:** In 1988, Sirovich and Kirby used linear algebra to overcome the problem of facial recognition. It started as a search for lower dimensional search for facial recognition. This method came into existence as Eigen faces. They came into a conclusion that feature extraction helps better in facial recognition. In 1991, Turk and Pentland conducted further research to find faces in image.

**Social media:** Facebook used conventional facial recognition system to group faces based on the person. This method uses deep learning but not convention neural networks. After this there is a certain development in the facial recognition system.

Recently Apple introduced face-Id as a biometric system which scans the 3d image and compares them.

1. **STEPS INVOLVED:**

The Convolutional face recognition has 4 steps in order to obtain results.

1. Face recognition

2. Face alignment

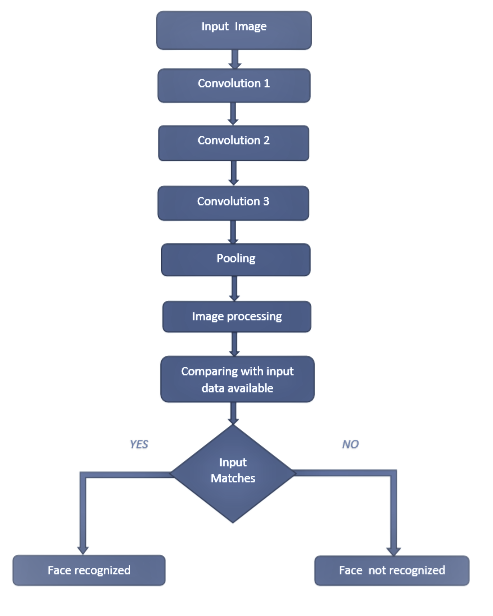
3. Feature Extraction

4. Classification.

In some constrained environments handcrafted methods like Local Binary Patterns (LBP) and Local Phase Quantization (LPQ) have achieved respectable facial recognition results. But due to variations in pose, illuminations, expressions the performance of these methods drop drastically in unconstrained environments. To obtain face recognition in unconstrained environment there is no robust feature. In the past years, Deep Learning which is generally called as CNN produced better results in unconstrained environment. Notably, the top three rates of facial recognition in unconstrained environment (FRUE) are obtained by CNNs.

1. **PROPOSED TECHNOLOGY**

**FLOW CHART:**



The main theme of this paper is face recognition using deep Learning. In this we are using Convolutional Neural Networks this is one of the most popular deep learning architecture. People are showing more interest in Deep learning because of its effectiveness and popularity. In this first there will be an input image with which we are working. We will perform a series of convolution then followed by Pooling.

**CONVOLUTION:**

The main block of CNN is convolutional layer; convolution in mathematical terms is merging the two information sets, and where as coming to our case convolution will be applied on the input with the help of the convolution filter to produce a feature map. One of the important point is that wee will perform the multiple convolutions by using different types of filters which results in different feature maps. Let us assume that we have a image of size 32\*32\*32 and the filter of size 5\*5\*3 if we carefully observe the depth of the filter and the image matches. Now the convolution can be performed by sliding the filter on the input image . We have to use different filters for different convolutions which results in different feature maps

**POOLING:**

The second step we perform here is pooling this is done to reduce the dimensions. Pooling layer will down sample each feature map and it reduces the height and the width keeping the depth same. There are different types of pooling out of which we have selected the max pooling. In this it will select the maximum value from each feature map and so the size of the image is reduced.

In the next step face extraction is done and next the image is processed and next the input image is compared with the datasets.

So with the help of deep learning that is by performing series of convolution and pooling we can recognize the face.

**ALGORITHMS USED:**

We are defining two approaches in this paper the first one is the Convolution neural network and the second is based on two models

**VGG-FACE NETWORK:**

This network contains sixteen CNN layers and 3 fully connected layers and five max pooling layers. This takes the input image and utilizes regularization in the fully connected networks. We have evaluated this method based on the LFW datafile and we have achieved an accuracy of 97.66%

**LIGHTENED CNN**:

Mostly CNN is used for the facial recognition techniques. Here we introduce the maxout concept from fully connected layer to convolution layer ,this leads to a new function named “Max Feature Map”(MFM). If we compare this with ReLU MFM can catch competitive information and compact representation simultaneously. We have evaluated this method based on the LFW datafile and we have achieved an accuracy of 96.23%

With the help of the above mentioned models we are recognizing the face.

1. **OBSERVATIONS:**

We summarized the main outcomes and observations below:

1) We conducted and evaluated the performance of this deep learning method in different conditions like light differences, pose differences, and misalignment of the pictures. Infact all other proposed theories based on the deep learning like Deep-face, Face-Net, VGG Face, Face-Id are trained using LFW and YouTube datasets whose performances are not assessed yet.

2) We found that even though deep learning representation works best for the facial recognition, it is not possible to find state of art results against pose and lighting. Hence, to improve results these factors should be taken into consideration during the time of pre training.

3) We found that this deep learning model is robust to the feature like misalignment and can reduce errors up to 10% of the intraocular distance.

4) VGG face model is better portable and reproductible than the Lightened CNN model. We also believe that there should contain further research on this method for better efficiency.

**COMPUTATION ACCURACY TRADE OFF:**

Fig 1.Flops compared with accuracy trade off

The above figure shows the accuracy trade off vs the no of flops. In this paper we are mainly focusing ion four models.

|  |  |
| --- | --- |
| architecture | VAL |
| NN1 (Zeiler&Fergus 220×220)  NN2 (Inception 224×224)  NN3 (Inception 160×160)  NN4 (Inception 96×96)  NNS1 (mini Inception 165×165)  NNS2 (tiny Inception 140×116) | 87*.*9% ± 1*.*9  89*.*4% ± 1*.*6  88*.*3% ± 1*.*7  82*.*0% ± 2*.*3  82*.*4% ± 2*.*4  51*.*9% ± 2*.*9 |

Fig 2.Architecture of networks

The table above thinks about the presentation of our prototypical structures on the holdout test set. Publicized is mean approval rate VAL at the “10E-3” bogus acknowledge rate.

Showing the FLOPS within the graph using x-pivot and the precision at “0.001” bogus acknowledge rate (FAR) on our client named “test-informational index”. This is intriguing to look at the solid relationship between the calculation, a model requires and the accuracy it achieves. The figure structures the five models (NN1, NN2, NN3, NNS1, NNS2) that we are going to examine furthermore.

Fig 3. Effect of CNN model

**Comparison table**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Existing System** | **Proposed System** |
| Accuracy | 93.16 | 94.02 |
| No. of Layers | 12 | 16 |
| Recognition time | Takes more time | Takes less time |

1. **RESULTS**

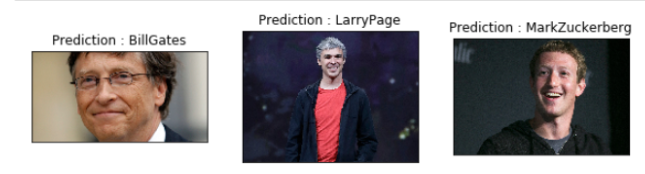
Images we have in the database



Testing images



Results



1. **CONCLUSION:**

We proposed to assemble a superior, adaptable, nimble, and ease facial recognition framework. We differentiated the proposed approach into a few smaller sub-ventures. In the first place, we considered the neural system and convolutional neural system. In the way of obtaining a profound learning method, we used the Siamese system which will prepare the neural networks accordingly. At that point we look at and analyze the accessible open-source informational collection, we picked the ORL dataset and prepared the model using GPU. This model will take a human picture and convert it to a vector. Similarly many vectors are created and these are contrasted with one another to find they both are of the same person.

1. **REFERENCES**

[1]  Y. Bengio, J. Louradour, R. Collobert, and J. Weston.Curriculum learning. In Proc. of ICML, New York, NY, USA, 2009. 2

[2] D.Chen, X.Cao, L.Wang, F.Wen, and J. Sun. Bayesian face revisited: A joint formulation. In Proc. ECCV, 2012. 2

[3]  D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In Proc. ECCV, 2014. 7

[4]  J. Dean, G. Corrado, R. Monga, K.Chen, M.Devin, M.Mao, M.Ranzato, A.Senior,P.Tuck-er, K.Yang, Q.V.Le, and A.Y.Ng. Large scale distributed deep networks.In P.Bartlett, F.Pereira, C.Burges,L.Bottou, and K. Weinberger, editors, NIPS, pages 1232–1240. 2012. 10

[5] J. Duchi,E.Hazan,andY.Singer. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 12:2121–2159, July 2011. 4

[6]  I. J. Goodfellow,D.Warde farley, M. Mirza, A. Courville, and Y. Bengio. Maxout netwo-rks. In In ICML, 2013. 4

[7]  G.B.Huang, M.Ramesh, T.Berg, and E. Learned Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environ-ments. Technical Report 0749,University of Massachusetts, Amherst, October 2007. 5

[8] Y. LeCun, B.Boser, J. S. Denker, D. Henderson, R. E.Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation,1(4):541–551, Dec. 1989. 2, 4

**Author’s Publication:**

