

American International University-Bangladesh (AIUB)

# Attendance Monitoring System by Face Recognition using Deep Learning Technology and Computer Vision

***Transfer Learning and vgg16 Pretrained Model***

|  |  |
| --- | --- |
| IHSAN MD. AL-AQIB | (18-37074-1) |
| MD. SAKIBUL ALAM UTCHAS | (18-37079-1) |
| S.M. IRAMUL MURSHED | (18-37069-1) |
| HASEBUL HASAN SHIHAB | (18-37031-1) |

*A Thesis submitted for the degree of Bachelor of Science (BSc) in Computer Science and Engineering (CSE) at*

*American International University Bangladesh in April, 2022*

Faculty of Science and Technology (FST)

## Abstract

Face recognition is a key component of human-computer interaction, and it's frequently employed in access control, monitoring, and identity verification systems. Face images vary with expressions, ages, and poses of people, as well as lighting conditions, therefore the face images of the same sample may be different, making face recognition difficult. Deep learning-based algorithms have outperformed classic machine-learning approaches in terms of accuracy and processing speed in image recognition. We applied transfer learning for Face Recognition in this paper to accomplish our goal of making attendance system. A dataset with 430 photos from 43 different classes was used. The weights and biases were fixed on the convolutional layer of the modified pre-train VGG-16, then the model was trained using the custom build dataset. The training and validation accuracy are both moderate, according to the results of the experiment.

## Declaration by author

This thesis is composed of our original work, and contains no material previously published or written by another person except where due reference has been made in the text. We have clearly stated the contribution of others to our thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support and any other original research work used or reported in our thesis. The content of our thesis is the result of work we have carried out since the commencement of Thesis.

We acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate we have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.



. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**IHSAN MD. AL-AQIB**

18-37074-1

Computer Science & Engineering



. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**S.M. IRAMUL MURSHED**

18-37069-1

Computer Science & Engineering



. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**MD. SAKIBUL ALAM UTCHAS**

18-37079-1

Computer Science & Engineering



. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**HASEBUL HASAN SHIHAB**

18-37031-1

Computer Science & Engineering

## Approval

The thesis titled **“Attendance Monitoring System by Face Recognition using Deep Learning Technology and Computer Vision”** has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science on **(April 30, 2022)** and has been accepted as satisfactory.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

#### Dr. Debajyoti Karmaker

*Associate Professor*

Department of Computer Science

American International University-Bangladesh

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

#### DR. KAMRUDDIN MD. NUR

*Associate Professor*

Department of Computer Science

American International University-Bangladesh

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

#### DR. MD. ABDULLAH - AL - JUBAIR

*Assistant Professor & Head-In-Charge [Undergraduate Program]*

Department of Computer Science

American International University-Bangladesh

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

#### Professor Dr. Tafazzal Hossain

*Dean*

Faculty of Science & Information Technology American International University-Bangladesh

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

#### Dr. Carmen Z. Lamagna

Vice Chancellor

American International University-Bangladesh

## Contributions by authors to the thesis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **IHSAN MD. AL-AQIB** | **MD. SAKIBUL ALAM UTCHAS** | **S.M. IRAMUL MURSHED** | **HASEBUL HASAN SHIHAB** | **Contribution (%)** |
|  | *18-37074-1* | *18-37079-1* | *18-37069-1* | *18-37031-1* |
| Conceptualization | 25(%) | 25(%) | 25(%) | 25(%) | 100 % |
| Data curation | 20(%) | 40(%) | 20(%) | 20(%) | 100 % |
| Formal analysis | 30(%) | 30(%) | 20(%) | 20(%) | 100 % |
| Investigation | 20(%) | 50(%) | 20(%) | 10(%) | 100 % |
| Methodology | 25(%) | 25(%) | 25(%) | 25(%) | 100 % |
| Implementation | 25(%) | 15(%) | 25(%) | 35(%) | 100 % |
| Validation | 30(%) | 40(%) | 20(%) | 10(%) | 100 % |
| Theoretical derivations | 10(%) | 10(%) | 75(%) | 5(%) | 100 % |
| Preparation of figures | 70(%) | 20(%) | 5(%) | 5(%) | 100 % |
| Writing – original draft | 20(%) | 20(%) | 35(%) | 25(%) | 100 % |
| Writing – review & editing | 30(%) | 30(%) | 20(%) | 20(%) | 100 % |

## Acknowledgments

This thesis is our amiable effort and our supervisor’s inventiveness and continuous inspiration. First of all, we would like to be grateful to the Almighty Allah, who granted us the effort to work on this project for the last two semesters. Special thanks go to our honorable supervisor **Dr. Debajyoti Karmaker**, Associate Professor, American International University-Bangladesh (AIUB) for this expansive support, motivation, and resourceful censure. His excellent supervision and continuous push made this project possible. We are humbly grateful to him for giving us the opportunity to work with him. A special thank goes to **Dr. Kamruddin Md. Nur**, Associate Professor, AIUB, for his suggestions. We also send our thanks to our honorable Vice Chancellor, **Dr. Carmen Z. Lamagna**, our Dean, **Prof. Dr. Tafazzal Hossain**, our Associate Dean, **Mashiour Rahman**, our Director, **Dr. Dip Nandi**, and our Head of Department, **Dr. Md. Abdullah - Al - Jubair** for their constant inspiration and support. Finally, our humble greetings are for our respectable parents who made the opportunities for us to get educated and pursue our interests for making great for mankind and made us think out of the limitations.

## Keywords

computer vision, deep learning, attendance system, cnn, face recognition, transfer learning

Table of Contents

[Attendance Monitoring System by Face Recognition using Deep Learning Technology and Computer Vision i](#_Toc101990810)

[Abstract ii](#_Toc101990811)

[Declaration by author iii](#_Toc101990812)

[Approval iv](#_Toc101990813)

[Contributions by authors to the thesis v](#_Toc101990814)

[Acknowledgments vi](#_Toc101990815)

[Keywords vii](#_Toc101990816)

[List of Tables xi](#_Toc101990817)

[List of Abbreviations and Symbols xii](#_Toc101990818)

[Introduction 13](#_Toc101990819)

[1.1 Introduction to Convolutional Neural Network (CNN) 14](#_Toc101990820)

[1.1.1 Convolution Layer 14](#_Toc101990821)

[1.1.2 Pooling layer 14](#_Toc101990822)

[1.1.3 Activation layer 14](#_Toc101990823)

[1.1.4 Fully Connected 15](#_Toc101990824)

[1.1.5 Classification layer 15](#_Toc101990825)

[1.2 Computer vision 15](#_Toc101990826)

[1.3 Face recognition 15](#_Toc101990827)

[1.4 Deep learning 15](#_Toc101990828)

[1.5 Transfer Learning 15](#_Toc101990829)

[Literature review 16](#_Toc101990830)

[1.1 Paper (Lucena, 2017, July) 16](#_Toc101990831)

[1.2 Paper (Yin, 2019) 17](#_Toc101990832)

[1.3 Paper (Heidari, 2020) 19](#_Toc101990833)

[1.4 Paper (Zeng & Wang, 2018) 19](#_Toc101990834)

[**1.5** **Paper** (Zhang H. e., 2017) 22](#_Toc101990835)

[**1.6** **Paper** (Said, 2020) 23](#_Toc101990836)

[**1.7** **Paper** (Wang, 2017) 24](#_Toc101990837)

[**1.8** **Paper** (Li, 2017) 26](#_Toc101990838)

[**1.9** **Paper** ([1] Pranav, 2020) 28](#_Toc101990839)

[**1.10** **Paper** ([1] Nakada, 2017) 29](#_Toc101990840)

[**1.11** **Paper** (Zhiqi, 2021) 31](#_Toc101990841)

[**1.12** **Paper** ([1] Coşkun, 2017) 33](#_Toc101990842)

[**1.13** **Paper** ([1] Xu, 2020) 35](#_Toc101990843)

[**1.14** **Paper** (Younis, 2017) 38](#_Toc101990844)

[**1.15** **Paper** (Nandhini, 2019) 40](#_Toc101990845)

[**1.16** **Paper** (K. Yan, 2017) 41](#_Toc101990846)

[**1.17** **Paper** (Xi, 2016) 46](#_Toc101990847)

[**1.18** **Paper** (Kakarla, 2020) 49](#_Toc101990848)

[**1.19** **Paper** (S. Sharma, 2016) 51](#_Toc101990849)

[**1.20** **Paper** (K. Bong, 2018) 53](#_Toc101990850)

[**1.21** **Paper** (Teoh, 2021) 54](#_Toc101990851)

[**1.22** **Paper** (Zhang Y. e., 2016) 57](#_Toc101990852)

[**1.23** **Paper** (Kamencay, 2017) 60](#_Toc101990853)

[Methods 63](#_Toc101990854)

[3.1 Dataset Information 63](#_Toc101990855)

[3.2 Data Preprocessing 63](#_Toc101990856)

[3.3 Structure of Model 63](#_Toc101990857)

[Results or findings 68](#_Toc101990858)

[Discussion 71](#_Toc101990859)

[Conclusion 73](#_Toc101990860)

[6.1 Future scope 73](#_Toc101990861)

[Bibliography 74](#_Toc101990862)

**List of Figures**

[Figure 3. 1 VGG 16 Architecture 64](#_Toc101987899)

[Figure 3. 2 Removing FC layers from VGG16 64](#_Toc101987900)

[Figure 3. 3 Modified VGG16 model 65](#_Toc101987901)

[Figure 3. 4 Transfer learning workflow 66](#_Toc101987902)

[Figure 4. 1 Train and Validation Loss Graph 68](#_Toc101987904)

[Figure 4. 2 Train and Validation Accuracy Graph 69](#_Toc101987905)

# List of Tables

[Table 3. 1: Details of the Dataset 62](#_Toc101982063)

# List of Abbreviations and Symbols

Mention all the abbreviations and the different symbols that is used in this document.

Abbreviations

CNN Convolutional Neural Network

SVM Support Vector Machine

TP True Positive

TN True Negative

FP False Positive

FN False Negative

LR Learning Rate

LBP Local binary patterns

**Chapter 1**

# Introduction

The instructor taking pupils' attendance at the start of each school day, and often numerous times throughout the day, is a typical sight at schools, colleges, and educational institutions all over the world. Even in virtual classrooms, taking attendance is required at the beginning of each lesson. Even office workers face similar situations daily. Tracking student or employee attendance makes it easier to keep track of absenteeism and timeliness. However, there are several drawbacks to using a manual attendance system — It's a lengthy procedure, to say the least. There is a significant quantity of documentation involved. Reduces the teacher's classroom teaching time and productivity. When tabulating attendance, there is a lot of room for human mistakes. The system can be manipulated in a variety of ways by students. Manually tracking attendance has always been a difficult and time-consuming procedure. Face recognition can make things easier to overcome these shortcomings.

Face recognition is the ability of a computer system to detect and recognize human faces in photos or videos quickly and accurately. For enhancing the performance of face recognition, a variety of algorithms and strategies have been developed. Deep learning has recently received a lot of attention for computer vision applications. Multiple faces can be detected and recognized automatically and quickly by the human brain. However, on the level of the human brain, it is extremely impossible to perform all of the tough activities on a computer. Facial traits are extracted and implemented using efficient algorithms, with certain changes made to improve the existing method models.

Face detection entails searching the input image for any faces, after which image processing cleans up the facial image to make it easier to recognize.

Face Recognition - determining who a person is by comparing the detected and processed face to a database of recognized faces.

The distinction between face detection and recognition is that in detection, we only need to see if there is a face in the image, however in recognition, we need to know who it is. Face features are extracted and compared to similarly processed faces in the database.

As a result, the focus of this thesis is on overcoming the challenges of attendance monitoring system. In this project, we will:

* Design and develop a pretrained cnn model based on VGG16 for face recognition.

• Overcome time constraints, heavy workloads, proxy attendance, and other issues.

* Show an alternative technique to take attendance.
* Propose a way for efficiently managing the attendance system.

We strongly want to make a substantial contribution to the attendance monitoring system by overcoming the limitations of the traditional attendance system.

## 1.1 Introduction to Convolutional Neural Network (CNN)

Convolution neural network is one of the representative network structures in depth learning. Its weight-sharing network structure makes it more similar to the biological neural network, which reduces the complexity of the network model and reduces the number of weights. The advantage of this method is that the input of the network is more obvious when the input is multidimensional, so that the image can be directly input to the network, which avoids the complex feature extraction and data reconstruction in the traditional recognition algorithm. A product network is a multilayer perceptron specially designed to recognize two-dimensional shapes that are highly invariant to translational, scale, tilt, or other forms of deformation. A CNN architecture is formed by a stack of independent processing layers. CNNs have an input layer, and output layer, and hidden layers. The hidden layers usually consist of convolutional layers, activation layers, pooling layers, and fully connected layers. Convolutional layers apply a convolution operation to the input. This passes the information on to the next layer.

### 1.1.1 Convolution Layer

The convolution layer (CONV) which processes the received input data. The first layer of a convolutional neural network is always a convolutional layer. Convolutional layers apply a convolution operation to the input, passing the result to the next layer. A convolution converts all the pixels in its receptive field into a single value. For example, if convolution is applied to an image, it will decrease the image size as well as bring all the information in the field together into a single pixel. The final output of the convolutional layer is a vector. Depending on the sort of problem to solve and the features to learn, several types of Convolution could be used.

### 1.1.2 Pooling layer

Pooling layer is another building block of a CNN. The pooling layer is used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer. The pooling layer operates on each feature map independently. The most common approach used in pooling is max pooling.

### 1.1.3 Activation layer

The activation layer, often misused as 'ReLU' with reference to the activation function (Rectification Linear Unit). Activation functions are a critical part of the design of a neural network. The choice of activation layer in the hidden layer will control how well the network model learns the training dataset. The primary role of the Activation Layer is to transform the summed weighted input from the node into an output value to be fed to the next hidden layer or as output. The choice of activation layer in the output layer will define the type of predictions the model can make.

### 1.1.4 Fully Connected

The “Fully Connected” (FC) layer, which is a perceptron type layer. Fully Connected Layer is simply, feed forward neural networks. Fully Connected Layers form the last few layers in the network. The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.

### 1.1.5 Classification layer

The classification layer (Softmax) that predicts the class of the input image. The Softmax classifier uses the cross-entropy loss. The Softmax classifier gets its name from the softmax function, which is used to squash the raw class scores into normalized positive values that sum to one, so that the cross-entropy loss can be applied.

## 1.2 Computer vision

Computer vision is a branch of artificial intelligence (AI) that allows computers and systems to extract meaningful information from digital images, videos, and other visual inputs — and then act or make recommendations based on that information.

## 1.3 Face recognition

A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces. The generic face recognition procedure conducts three primary processes, which are:

(i) face detection

(ii) feature extraction

(iii) face recognition

## 1.4 Deep learning

Deep learning is a subset of machine learning, which is essentially a three- or more-layered neural network. These neural networks attempt to simulate the behavior of the human brain, albeit with limited success, allowing it to "learn" from large amounts of data.

## 1.5 Transfer Learning

Local Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

**Chapter 2**

# Literature review

## Paper (Lucena, 2017, July)

Transfer learning of a pre-trained CNN is used in this approach. The first technique uses the source model as an "off-the-shelf" feature extractor, using the output of a chosen layer in the target model, which is the only one trained for the new task. A more advanced approach uses backpropagation to "fine-tune" the source model, retraining its weights in whole or in part. Transfer learning can be used to avoid overfitting a large network if there is insufficient data to train it from scratch. It can also save computational resources, as training from scratch can take days to weeks. In this case, the authors chose the CNN architecture VGG-16, which was pre-trained on the ImageNet database, and its team secured first and second place in the ImageNet ILSVRC-2014 localization and classification tasks. They used fine-tuning as a transfer learning approach.

Except for the top layers, the proposed face anti-spoofing network (FASNet) follows the VGG-16 architecture. The code for FASNet can be found at https://github.com/OeslleLucena/FASNet. The VGG-16 architecture is a 2D CNN with an input size of 224 × 224 pixels. It has 16 convolutional layers in total, with 64 filters in the first block, 128 filters in the second, 256 filters in the third, and 512 filters in the fourth and fifth blocks. A kernel of size 33 is used in each convolution. All max-pooling layers are run in a 22-second window with a stride of 2. The activation functions are linear rectifier units (ReLU).There are three fully connected (FC) layers, the sizes of each one are 4096, 1000, and 1000. Only the top layers of the FASNet vary from the VGG-16. One of the FC layers was removed, and the other two were reduced in size to 256 and 1. Furthermore, the Adam optimizer is used with the paper's configurations, changing the learning rate to 104 and the weight decay to 106. In addition, the decision function has been changed from a softmax to a sigmoid, which is often more effective for binary classification.

This database, known as 3DMAD, contains 76,500 frames of 17 people in 255 videos (300 frames per video) recorded with a Kinect for both genuine subjects and mask attacks. A depth image, a corresponding RGB image, and manually annotated eye positions make up each frame. Only color images were used for the requirement. All of the data is divided into three sessions, with the first two sessions containing only genuine subjects and the third session containing mask attacks. The first two sessions each have seven videos, while the third session only has five.

REPLAY-ATTACK is a database that contains 1,300 video clips of 50 subjects attempting photo and video attacks. This database's subjects were gathered under two different lighting conditions: controlled and adverse. The first condition had the office light on, the blinds were drawn, and homogeneous background. The second condition had the following characteristics: the blinds were up, the background was complex, and the office lights were turned off. In addition, attack protocols are classified based on the type of device that was used to launch the attack, such as print, mobile (phone), and high-definition (tablet).

Each face anti-spoofing database provided test folders for the experiments, which were conducted on the train. Because the architecture is 2D and the public databases are composed of video clips, some preprocessing was required. The weights were frozen from the bottom layers up to the third block of the face anti-spoofing network (FASNet), and then fine-tuned weights from the fourth block up to the top layers via backpropagation. After that, the method was evaluated on the REPLAY-ATTACK and 3DMAD test folders. The algorithm was implemented with the Keras library and Theano as the backend. Furthermore, all analyses were carried out on a computer with an Intel (R) Xeon (R) E5506 2.13 GHz processor and 6 GB of RAM, as well as an NVIDIA Tesla K40c GPU (12 GB).

There were two preprocessing steps used: (1) subsampling and (2) face detection. The first step was to extract half of each video's frames per second. In step two, the OpenFace face detector algorithm was used to find the region of interest (ROI) corresponding to a face. Then, using OpenFace algorithms, the image is cropped to a 96-pixel window and aligned with the faces to the center based on the position of the nose and eyes. The assessment was based on two metrics: accuracy (ACC) and half total error rate (HTER), both of which are commonly used to evaluate biometrics systems. Because the CNN output scores are probabilities, HTER was computed assuming a value of 0.5, and its estimation was not required.

Six other approaches were compared to the method, three of which were based on traditional machine learning algorithms, one on shallow neural networks, and the other two on CNNs. For all of the conventional machine learning methods submitted to the 3DMAD and REPLAY-ATTACK databases, this method outperformed HTER. FASNet outperformed almost all state-of-the-art methods on the REPLAY-ATTACK benchmark, losing only to the Multi-cue Integration approach, and this method achieved an HTER of 0.00 percent on the 3DMAD database. It's worth noting that the proposed method only uses static features, whereas the Multi-cues Integration method, for example, uses both static and dynamic features.

In this paper, a new approach to face anti-spoofing methods is presented that employs transfer learning in convolutional neural networks. On the REPLAY-ATTACK and 3DMAD databases, the HTER was found to be 1.20 percent and 0.00 percent, respectively, outperforming almost all state-of-the-art methods.

## Paper (Yin, 2019)

Face recognition is a consistent success story for deep learning, but tactics for extracting value from enormous volumes of diverse data are mostly unexplored. With only a few samples available for each class, a considerable fraction of the data is under-represented (UR). In the long term, classifiers that ignore this UR data are more likely to be prejudiced. About 39% of the 10K participants have fewer than 20 images, resulting in insufficient training data. While sampling UR classes more frequently alleviates the problem, biased choice limits still arise.

In this paper, the authors propose to use Feature Transfer Learning (FTL) to create less biased face recognition classifiers. UR classes have their feature distribution matched to that of ordinary classes. Transferred data should not be used for training since the distributions of the classes may be distorted. This study makes advances to the field of facial recognition theory by presenting a unique feature transfer approach and a simple but successful metric regularization (FTL). Ablation experiments on LFW and IJB-A have shown that our FTL design is effective.

One-shot learning tries to recognize a picture for a certain class based on a small number of training photographs. Nonparametric classification approaches are applied in certain circumstances, whereas strong regularization is imposed in others. Other generative model-based approaches, such as attribute-guided feature descriptors, have been studied in recent years.

**Feature transfer learning:** Feature transfer learning is the process of transferring information from one domain to another. Additional monitoring is not included in our approach since it may introduce new bias. The researchers depict intra-class variance using a parametric technique, assuming that regular and UR classes have the same feature variance distribution.

The Proposed Approach contains our recommended metric regularization, feature transfer framework, and alternate training strategy as part of the recognition backbone architecture. In Section, we look at the problems that emerge while using UR classes for face recognition training, as well as how to address them. The rich-feature layer's purpose is to improve intra-class variance in UR classes at a lower level. An encoder harvests valuable data from an input image and reconstructs it with the help of a decoder Dec. Then, via filtering layers, a better discriminative representation will be learned. The identity-specific feature center representation of regular classes removes non-identity elements including posture, expression, and lighting. Due to a dearth of data, UR class center estimate is frequently skewed towards identity-irrelevant characteristics such as posture.

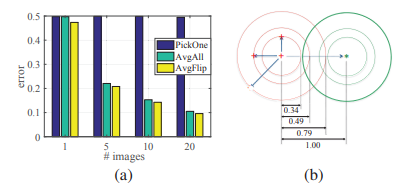


Figure 1. are averaged across 1K classes. Circles from small to large show the minimum, mean and maximum distances from within a class to outside the center.

The feature center evaluation is a crucial phase in the feature transfer process. We choose a subset of 1, 5, 10, and 20 images at random to recreate a UR class. As seen in Figure 2, using our "Avg Flip" method reduces inaccuracy. In comparison to intra-class variance, the error is minor.

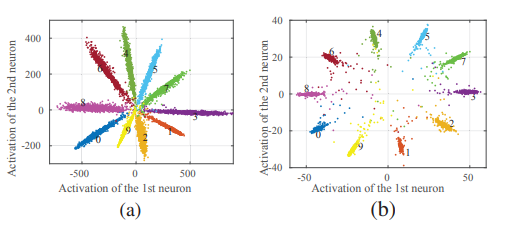


Figure 2. Toy example on MNIST to show the effectiveness of our m-L2 regularization. Figure shows the feature distributions for models trained without (a) and with (b) m-L 2 regularization, as well as those trained with and without it.

The m-L2 regularization is a generic regularization that may be simply applied to various recognition systems. Soft max loss can be reduced by increasing the feature norm, however this may lead to over-fitting. Models trained with m-l2 increase softmax from 99.06 % to 99.35 % on the MNIST testing set.

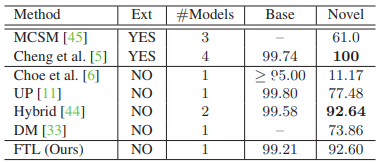


Table 1 shows the results of a one-shot learning exercise. On base classes, the result is reported as rank-1 accuracy, whereas on novel classes, it is reported as Coverage @ Precision = 0.99. "External Data" is what "Ext" stands for.

Table 1 shows that utilizing the output from the softmax layer as the confidence score, we can achieve 92.60 % coverage at a precision of 0.99 using single-model single-crop testing. Both methodologies make use of model ensemble and multi-crop testing. When compared to techniques with equivalent parameters, we achieve competitive performance on the base classes and 15% higher accuracy on the new classes.

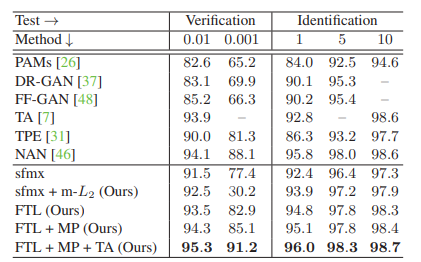


Table 2 shows the results of face recognition on the IJB-A. "MP" and "TA" stand for media pooling and template adaptation, respectively. The results of verification and identification are given at various FARs and grades.

With margins ranging from 0.6 % to 2.8 %, our FTL greatly enhances the performance. We combine our suggested technique (FTL + MP + TA) with media pooling (MP) and template adaptation (TA) to get consistently improved outcomes.

## Paper (Heidari, 2020)

A siamese network is a neural network architecture featuring two parallel neural networks that are comparable. The weights are shared between the two networks, and the networks have identical configurations with similar weights and parameters. The outputs of each network are merged to offer some predictions. Each network has a different input (picture).

In this paper transfer learning was utilized. Transfer learning was used in this work. To extract features in siamese architecture and fine-tune it, a pre-trained VGG-16 CNN model was utilized, which was pre-trained using the ImageNet dataset and achieved top 5 accuracy of 92.7%. In fine tuning part, all the fully connected layers of VGG-16 were eliminated. Then three fully connected layers was added which contain 512 neurons each and RELU is used as ab activation function. Except for the block-5 levels, which include three convolutional layers and one pooling layer, all convolutional layers were frozen. Also, the input size of this network was set to 128 × 128 × 3. To train the proposed model “contrastive loss Function” was used.

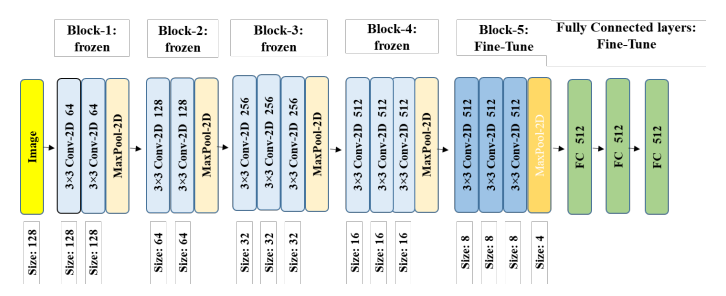


Figure: VGG-16 network architecture with fully connected layers on top for fine-tuning

To train the model LFW was used which contains over 13000 images of 1680 classes with image size of 250 × 250 were collected from web pages. To give input to the siamese network 15000 pair of images and the labels was generated and from that 60% used for training and 40% used for validation.

After training this model was achieved 95.62 ± 0.42 % accuracy.

## Paper (Zeng & Wang, 2018)

Face recognition (FR) with single sample per person (SSPP) is a challenge in computer vision, as shown in paper (Zeng & Wang, 2018) Because there is only one sample to train, it is difficult to predict facial variation such as pose, illumination, and disguise. TDL achieves state-of-the-art performance in SSPP FR using AR face database, Extend Yale face database, FERET face database, and LFW database.

In this paper (Zeng & Wang, 2018), a scheme is proposed that combines traditional and deep learning (TDL) methods to process the task by expanding the sampling method and a well-trained deep convolutional neural network (DCNN) model to overcome the lack of training sample in SSPP FR. It first learns an intraclass variation set, which is then added to a single sample in order to expand the sample. The following are the steps for creating an intraclass variation set. Create intraclass variation images based on images from an extra frontal face dataset first. Assume that an extra frontal face dataset contains m subjects, each of whom has (n−1) variation images and one neutral image, allowing us to express the dataset using X.let Xij represent the ith person’s jth variation image, where i ∈ [1, m], j ∈ [1, n], and let j = 1 represent the neutral face. We use variation image of the database (Xij, j≠1) minus its corresponding neutral image (Xi1); As a result, we get the following variance of the variation image relative to its neutral image:

εij = Xij−Xi1, j≠1, (1)

which represents the jth intraclass variation image of the ith subject in relation to its neutral image. Then, to reduce the intraclass variation image error, find the average intraclass variation image that has the same variation as these intraclass variation images, as follows:

j = (2)

Finally, in the forward step, creating an intraclass variation set based on the average intraclass variation images learned. The following is a representation of the situation: (3)

The framework for producing intraclass variation sets is shown. The face image is then detected and cropped from the new input face image using C++ and MATLAB, and the face image is then resized to the same size using the intraclass variation set. Finally, the intraclass variation set is applied to the aligned face image for image expansion:

where Xk1 represents the neutral face image of the person k and Dek represents the person k's expanding samples. A single sample is expanded to many samples using this method.

The second method follows the first. It is difficult to use DCNN in SSPP FR because it requires a large number of samples to be trained. To address this issue, it was began by introducing a well-trained DCNN using transfer learning. The learned DCNN is then fine-tuned by selecting some expanding samples. Finally, to implement the experiment, the fine-tuned DCNN is used. The research is carried out with the help of a lightened CNN that can learn a compact embedding for face recognition. Max-FeatureMap is a new activated function in the lightened CNN that introduces max out in the fully connected layer to the convolution layer. Given an input convolution layer C ∈ Rh×w×2n, The activation function for the Max-Feature-Map can be written as follows::

where the channel of the input convolution layer is 2n,

i ∈ [1, h], j ∈ [1, w]

The CASIAWebFace database is used to train the lightened CNN. The CASIA-WebFace database has a total of 493,456 face images and contains 10,575 people. It is first preprocessed before being used to train the lightened CNN. The images are converted to grayscale and normalized to 144 × 144 as part of the preprocessing. It is then used to train the lightened CNN after it has been preprocessed. After completing all of the tests, it was discovered that the patch-based method TLC is extremely competitive in the AR face database, outperforming the generic learning method LGR by 1.7 percent, 2.1 percent, 2.5 percent, and 3.1 percent under various conditions, but the proposed TDL, on the other hand, outperforms TLC by 0.8 percent, 12.9 percent, 3.7 percent, and 7.4%, respectively. TDL's accuracies are particularly high under illumination, expression, and disguise conditions, reaching 100%. The Yale B face database contains 38 subjects, each with 64 images taken in various poses and lighting conditions. The direct method still has the lowest recognition rate, and the DL method SSAE outperforms it by 2.8 percent and 4.4 percent, respectively. However, the generic learning methods SVDL and LGR outperform SSAE by 2.8 percent and 4.4 percent, respectively. TDL, on the other hand, outperforms SVDL and LGR by 3.3 and 1.7 percent, respectively. Furthermore, the accuracy of the Extend Yale B face database is lower than that of the AR face database. The FERET face database contains 1400 images of 200 subjects in various poses, expressions, and lighting conditions. The direct method consistently outperforms other approaches. The results of the expanding sample method are also poor. SVD-LDA, an expanding sample method, outperforms PCA by 1.5 percent; however, SOM, the best direct method, outperforms SVD-LDA by 5.5 percent, However, DMMA, a patch-based method, outperforms SOM by 2%. TDL, the proposed method, achieves the best results and outperforms DMMA by 0.9 percent. The 0e LFW database contains 1680 subjects with over 13000 images, all of which were gathered from the internet and included many unconstrained conditions. All of the accuracies are extremely low, with none exceeding 31%; however, the proposed method TDL achieves the highest accuracy of 74 percent, outperforming the second LGR by 43.6 percent more than twice. The LFW database, in particular, is collected under unrestricted conditions. The experimental result shows that, while constrained images are used to create the intraclass variation set, they can also be used in unconstrained conditions.

In SSPP FR, TDL achieves state-of-the-art performance among these methods. Furthermore, this paper is the first to use DCNN in SSPP FR, allowing DCNN to be used in a single sample or a small number of samples.

* 1. **Paper** (Zhang H. e., 2017)

This paper (Zhang H. e., 2017) uses the Softmax regression method to classify and complete face recognition using the ORL, YALE, and FERET databases, and presents a face recognition algorithm based on a local binary model and deep convolution network. This paper uses the LBP features of a face image as an input to CNN, trains the CNN network with the LBP features, and then uses the trained network for face recognition to overcome the disadvantages of CNN gray scale instability and more effectively identify the trained CNN network.

In a 3x3 neighborhood, LBP is defined. The threshold is the pixel gray value in the center. Compare the value of the pixels that surround it by 8 pixels. If the value of the pixels around the center pixel is higher than the value of the surrounding pixels, Otherwise, the pixel position is marked with a 1; or else, it is marked with a 0. Finally, you'll get an 8-bit binary number. This point's LBP code can be found by converting it to a decimal number. To obtain the face feature, the image is reconstructed as a histogram. W is placed within an arbitrary radius of the circular neighborhood by the improved LBP algorithm. LBP operators are commonly used to select the number of pixels P to describe the image's characteristics, Expressed in LBPP,R . gi represents the pixel gray value in the radius's neighborhood, i=0,1,2......p-1,s(x) represents the center pixel gray value, and represents the Boolean value of the neighborhood pixels after comparing the gray values of the center point's pixel.

Three convolutions (conv1,conv2,conv3), two pooled layers (Pool1,Pool2), and a fully connected layer (conv1,conv2,conv3) make up the network structure. The base layer C2 and C3 have 3x3 convolution kernels, compared to 5x5. The latter two convolutions use 3x3 for better results. because the network's non-linear capabilities are increased by two 3x3. If L is a roll over layer, k represents the number of features generated by the L layer to the L+1 layer in Equation 2, where m represents the set of output patterns on the previous layer, xlj represents the output of the jth neuron after convolution

The activation function f is used in this paper, and it is the ReLU activation function. The 2x2 Max pooling method is used in this article; assume that down indicates a downsampling operation, multiplicative bias is represented by βlj. The following is the calculation expression for a feature graph in the pooling layer.

The classifier layer is the convolution neural network's output layer. In this paper, the Softmax regression classifier was used. For a specific training, enter the data x. The output category y belongs to one of the following classes: 1,2,......k. There are k classes in total, and this article is number ten. Assume the input data x has been specified. The probability distribution of its class y = i is as follows: e denotes the natural logarithm's base, ș i denotes the parameters to be fitted, and The transpose is indicated by the superscript T, The meaning of is the probability that the input data x corresponds to each class i, i can take 1 to k.

Face recognition results were 96.6 percent in the ORL database, 96.6 percent in the YALE database, and 95.6 percent in the FERET database. In comparison to the general deep convolution network algorithm, the algorithm presented in this paper reduces the cost of training the network and is more practical.

* 1. **Paper** (Said, 2020)

The face recognition model proposed in this paper (Said, 2020) is based on the LBP feature and CNN. The picture is first turned into an LBP feature map, which is then utilized as the input for CNN to train. When it comes to facial recognition, Then into CNN for identification.

The original LBP operator is described as a 3 \* 3 window with the window center pixel as a threshold, next to the 8 pixel gray value with its comparison, where the pixel is marked as 1 if the surrounding pixel value is larger than the center pixel value, otherwise it is 0. This method compares 8 points in the 3 \* 3 neighborhood to create an 8-bit binary number (which is commonly translated to LBP code, which is 256 decimal), and the LBP value of the window's center pixel is produced and reflected by this value. This area's texturing information. Obviously, the recovered LBP operator may obtain LBP "coding" at each pixel point, after which the original LBP operator is extracted for an image (the gray value of each pixel is recorded) (recording the LBP value of each pixel).

A 28 × 28 input layer, two convolutions, two pooling layers, and a fully linked output layer make up the neural network. The input 28 \* 28 picture is convoluted via the first convolution layer, resulting in six 24 \* 24 characteristic matrices, and then through the first layer of a pool convolution sampling feature, resulting in six 12 \* 12 characteristic matrices. The second convolution layer then generates a second 8 x 8 feature matrix, followed by sampling the second layer's convolution features to yield 12 x 4 feature matrices. Two neurons, the output vectors 2 \* 1, are fully coupled to the last 12 4 \* 4 feature matrices. In this model, the training data set was utilized to train the CNN to build the classifier, and the classification result was then tested using the test dataset. There are 500 positive samples and 2000 negative samples in the training dataset, whereas there are 200 positive samples and 400 negative samples in the test dataset.

Accuracy is defined as the capacity to judge the full sample set correctly, with the positive judgment being correct and the negative judgment being correct.

Accuracy = (TP + TN) / (TP + FN + FP + TN);

* Sensitivity: The ability of a positive sample to be predicted as a positive sample, Sensitivity = TP / (TP + FN);
* Specificity: The ability of negative samples to predict negative samples, Specificity = TN / (TN + FP);

The accuracy, sensitivity, and specificity of this improved model LBP+CNN are 95.33 %, 90.50 %, and 97.75 %, respectively, whereas without LBP simply CNN performs 91.83 %, 89.50 %, and 93.00 %. The effect of the proposed classification method is better than the effect of the former classification when the accuracy, sensitivity, and specificity of the two categories of classification are compared; according to the ROC curve and AUC analysis, the face recognition method based on LBP and CNN is superior to CNN face recognition.

* 1. **Paper** (Wang, 2017)

This paper (Wang, 2017) uses Convolutional Neural Networks to recognize faces in grayscale photos in a biometric system. It presents a Deep Learning model structure that can improve current state-of-the-art precision and processing time. Two convolution layers, a fully linked layer, and a classification layer make up the proposed network. An activation layer and a maxpooling layer follow each convolution layer. After each convolution layer, apply two regularization techniques: batch normalization and dropout. The suggested CNN's input image size was set to 32x32x1, the kernel size was set to 3x3, and the number of filters was set to 16.

Because the input picture is a grayscale image with fewer characteristics to learn, the number of filters was lowered. The activation layer's ReLU function. The kernel size was set to 2x2 for the max-pooling layers, and the stride was set to 2. The number of filters in the second convolution layer was increased to 32, and the same kernel size as the first convolution layer was employed. A batch normalizing approach was used after each block of convolution, activation, and max pooling. It is suggested that a dropout mechanism be included to avoid the overfitting problem. In order to obtain excellent performance, regularization methods, batch normalization, and dropout are only used during the training process. In order to summarize and aggregate learnt information, a fully linked layer was added to the network. To avoid the problem of overfitting, the dropout approach was used after the completely linked layer. The probabilities of each class were computed using the Softmax function as the output layer. All probability must add up to one. The Softmax is computed as (1):

When x is the input data, y j denotes the output of the neural network for class j, and w j, w i denotes the weight of the neuron at positions I j. The suggested CNN is shown in detail in Figure 1. According to the training dataset, which includes 40 classes to identify, the proposed neural network has 40 outputs.

The Tensorflow Deep Learning framework and the OpenCV library were used to create all of the algorithms, which were written in Python. The ORL dataset was utilized to train the proposed CNN for facial recognition. The dataset comprises 40 different faces, each of which is classified into a separate class. Each class has ten pictures that are 92x112 pixels in size and have 256 grey levels per pixel. The photographs were divided into 40 distinct folders, each containing ten images from the same class. There are 400.pgm photos in the collection.

Faces in frontal view with an upright or modest left-right rotation may be seen in all of the photographs. Figure 3 depicts all of the ORL dataset's pictures. A training set and a testing set were created from the dataset. The training set included 6 photos for each class, while the testing set included 4 images. The suggested CNN uses the Categorical Crossentropy as a loss function. The Categorical Crossentropy can be computed as:

where y is the anticipated class and y is the desired class The Adam gradient descent approach was used to optimize the suggested loss function. The Adam algorithm's updated weights may be calculated as follows:

where , and presents the updated weights and the old weights, while is the bias corrected first moment and is the the bias corrected second moment.

The proposed CNN was trained for 20 epochs using the ORL dataset, with 400 iterations per epoch. Figure 4 shows the loss optimization as well as the accuracy curves. On the testing set, the loss function achieves a minimum of 0.12. (validation). The suggested CNN achieves 99.78 % training accuracy, 98.7 % validation accuracy, and 231 frames per second inference speed. The suggested CNN achieves a high level of accuracy.

* 1. **Paper** (Li, 2017)

FaceNet network topologies were used to develop a facial recognition system based on OpenFace in this paper (Li, 2017). Unlike other systems, we propose an intelligent model training method called S-DDL (self detection, decision, and learning) that uses an incremental SVM algorithm to allow our system to update the classification model in real time while running. The S-DDL approach may significantly enhance accuracy in a short period of time during system running, and the incremental SVM algorithm outperforms the classic one.

Face detection, face alignment, face feature extraction, and face classification are the four key processes in the face recognition process. We start by locating the biggest face in each image and returning the bounding box around it. Then, for each face, we utilize the landmark with 68 key points to align the nose and eyes with the mean placements. To make our model more efficient in the training phase, we take the biggest face from each image and align it with a size of 96\*96. This method extracts features using OpenFace, which is based on FaceNet. The softmax layer was replaced with an incremental classifier.

This method can distinguish fresh photos taken from web cams using the present classification model. The newly uploaded photographs will be tagged when these images are saved in the train set. When the system detects that the number of newly uploaded photos has reached a specified threshold, the incremental SVM training technique is used to build a new SVM classification model based on the previous model and fresh data. The classification model will then be instantly updated, and the accuracy will be calculated. This algorithm can achieve self-detection, self-decision, and self-learning throughout system operation in this fashion.

Here's how the algorithm works: Consider a dataset D; the KKT conditions on the point x mC ij split data D into three groups based on the value of g mij for all i=1,..,K,j-i+1,..,K, and C ≥ 0 regulates the amount of outliers and corresponds to assigning a bigger penalty to errors when C is greater:

Support vectors S are on the border, error vectors E are greater than the margin, and data vectors D are within the boundary.

When a new data set x c is introduced, the coefficient \_cpq=0,p=1,...,K,q=p+1,...,K change value progressively, and the parameters of the current support vectors are changed to meet the KKT criteria. When fresh data x c is added, the adaption of g mij can be stated in numerous ways, for example:

where is the coefficient to be incremented and is the kernel function of the SVM, so that . Coefficient are defined.

From equation (1), for all the support vectors , we get , then . Therefore equation(2) and (3) can be written as the following matrix equation:   
Where , and expresses the weights of support vectors sv nij in the C i class

The classifier was trained using 26 IDs, each with 300 photos. When the resolution exceeds 640\*360, the precision remains relatively constant. Based on the results, we decided to set the input picture resolution at 640\*360, which ensures the system's high precision and real-time. The results imply that using incremental SVM to improve face recognition performance is beneficial. While incremental SVM training takes less time, classical training takes much longer.

* 1. **Paper** ([1] Pranav, 2020)

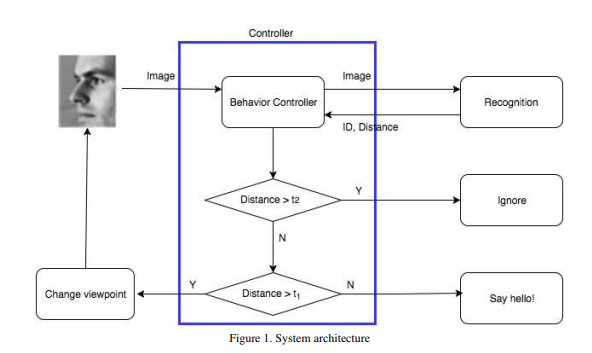
The majority of face recognition systems are made up of two key modules: feature extraction and classifier. Convolutional Neural Network (CNN) is a deep learning technique that is typically recommended for image-based applications. In this study, the design of a real-time face recognition system using CNN is proposed, followed by an evaluation of the system based on modifying the CNN parameters to improve recognition accuracy.

The proposed CNN architecture is built with Keras, an open-source neural network toolkit that runs on top of TensorFlow. The CONV layer contains the CONV and RELU layers. The real-time input image from the camera is first fed into the Viola Jones algorithm for face detection, and then the cropped face image is converted into gray scale, scaled to 120×120 pixels, and fed into the first convolution layer, which consists of 32 filters of size 3×3 pixels. The first's output is sent to the second CONV+RELU layer, which has a different set of 32 filters of size 3×3 pixels. The output of the second CONV+RELU layer is sent to the POOL layer, which uses the max pooling function and a window size of 4×4 pixels. During the assessment, it was discovered that max pooling delivered superior accuracy than average pooling for the suggested job, so max pooling is used in this study. The output of the POOL layer is routed to the DROPOUT layer. Dropout rates of 0.1, 0.5, and 0.8 were tested, and during evaluation, a dropout rate of 0.5 produced the highest accuracy for the proposed application. As a result, a drop rate of 0.5 is used. Following levels of CONV+RELU, POOL, and DROPOUT are not provided because there isn't much information left in the DROPOUT layer output. The output is then flattened before being sent to the DENSE/FC layer for classification. Because the proposed real-time system is currently intended to identify faces of five people, the final DENSE layer is 5×1, but the system created using AT&T datasets includes a DENSE layer of size 40×1 to classify faces of 40 people.

The performance of this model was evaluated using the AT&T database, which included 10 images from 40 different people. Out of 400 images, 320 were used for training (8 images from 40 individuals) and the remaining 80 for testing. The performance of the proposed system is evaluated by varying the number of filters in the convolution layer and the window size of the convolution filter for different pooling window sizes. It was found in this paper that a convolution filter of size 3×3 pixels with 32 filters achieved a maximum identification accuracy of 98.75 percent for the suggested system when a pooling window size of 2×2 and 4×4 pixels was used.

The performance of the proposed system is also evaluated for real-time inputs via camera. The samples of 5 people, including authors and family members, are used to evaluate the proposed real-time system, and 40 photos of each participant are taken. Twenty images from five different people were used for training, and the remaining 100 were used for testing. It was discovered that a maximum recognition accuracy of 98.00 percent was obtained for a real-time system using 32 convolution filters with pooling window sizes of 2×2, 3×3, and 4×4 pixels and different convolution filter window sizes.

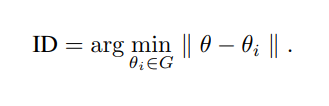
* 1. **Paper** ([1] Nakada, 2017)

AcFR is an active face recognition system that employs a convolutional neural network and operates in accordance with human behavior in typical face recognition situations. The recognition module extracts facial image characteristics using a pre-trained VGG-Face CNN and a nearest-neighbor identity recognition criterion. The system architecture is represented in the diagram below.

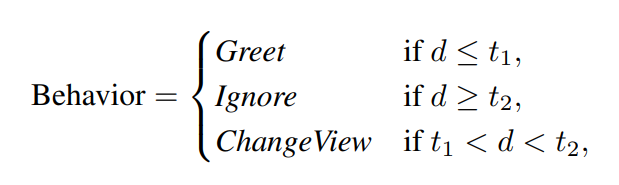
The AcFR system is made up of two main components: a behavior controller module and a face recognition module. Detection, alignment, representation, and classification are the four stages of the current face recognition pipeline. Using classification methods and a good face representation, the system can predict identification. Faces are recognized in this project using the technique developed by Mathias et a.

The VGG-Face network, a 16-layer CNN, was deployed and trained on more than 2 million celebrity photos in this experiment. The Caffe [5] Zoo Library includes a pre-trained VGG-Face model, making it simple to use. Researchers can extract acceptable image characteristics from the output of the fc-6 layer using VGG-Face and apply them in the subsequent classification step. A 4,096-dimensional feature vector is used to represent each 224×224 image in this method.

Researchers experimented with a variety of categorization techniques. With k = 30, the KNN classification achieved 90% accuracy. As a result, they decided to use the Nearest-Neighbor (NN) classifier. Given the feature vector associated with an unknown image, NN will calculate its Euclidean distance from each of the feature vectors I stored in the gallery G and return the image's identification as



They also employed Euclidean distance in the ensuing behavioral model.

For the purpose of behavior modeling The controller module is set up with two distance thresholds, t1 and t2. Given the output of the recognition module, the controller can simulate the aforementioned behaviors in the following definite way. 

Where d = minθi∈G || θ - θi ||. In the first and second circumstances, the system feels certain that the subject is a friend or a stranger, respectively, whereas of the third case it must learn further information via a change in viewpoint.

Caffe was utilized to create our recognition module. The project built in Elastic Compute Cloud (EC2), which provides a scalable computing capability on Amazon Web Service.

The PIE facial image library utilized in this study contains 41,368 photos of 68 persons captured under 43 distinct lighting circumstances, 4 different face expressions, and 13 imaging views ranging from -90◦ to 90◦.

To evaluate the performance of the recognition module, one frontal image of each person was reserved and saved in the gallery with its VGG-Face feature vector, while the remaining photographs were used for testing. The accuracy and distance are, as expected, view-dependent. The accuracy of photos with an angle closer to the frontal view can reach 100 percent, and the distance is also reduced. To demonstrate how the AcFR system works when monitoring strangers, 10 people were chosen at random from the gallery and made strangers. The average distances under different viewpoints for these strangers range from 286 to 350, which is close to the distances of the worst views at -90 and 90. This demonstrates that the system can distinguish between strangers and friends. As a result, given appropriately configured thresholds in the behavior controller module, it demonstrates appropriate behavioral modeling.

The AcFR system takes approximately 2.2 seconds to initialize, and each image is recognized in approximately 0.067 seconds. The AcFR system takes about 2.2 seconds to initialize, and each image is identified in about 0.067 seconds.

* 1. **Paper** (Zhiqi, 2021)

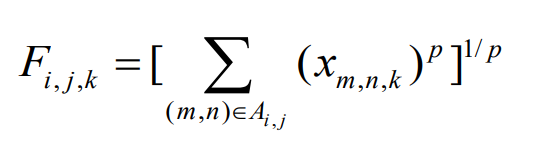
VGGNet is a network structure proposed by Simonyan K, Zisserman A, and the Oxford University VGG visual geometry group. It is based on the LeNet and AlexNet frameworks, employs 19 layers of network depth, and ranks first in the 2014 SVILRC and second in the classification. Massive training samples and high-performance computers are required for these networks. Many well-known convolutional neural networks currently rely on massive training samples and high-performance computers. To address this issue, this paper enhances the deep learning algorithm VGGNet for image classification and proposes the MicroFace face recognition method.

VGG-16 configuration consists of 13 convolution network layers and 3 fully connected layers, of which the convolution network layers are all the same with 33 convolution cores, step length of one, and filling of one, and stacked sequentially in 2 or 3 convolution network layers to build a module. Each module is followed by a maximum pool size of 22, a step size of 2, and a Fill value of 0. It has three fully connected layers at the end of the structure, and the SoftMax layer outputs the identification class.

The first two fully connected layers were removed in order to reconstruct VGGNET. The full connection layer of VGG-16 has the most parameters, and reducing the connection layer is the best way to reduce network parameters. To reduce the parameters according to the convolutional neural networks DeepID and GoogLenet, the VGG-16 network's three full connection layers were combined into one fully connected layer. The number of training samples and calculation time are reduced as a result.

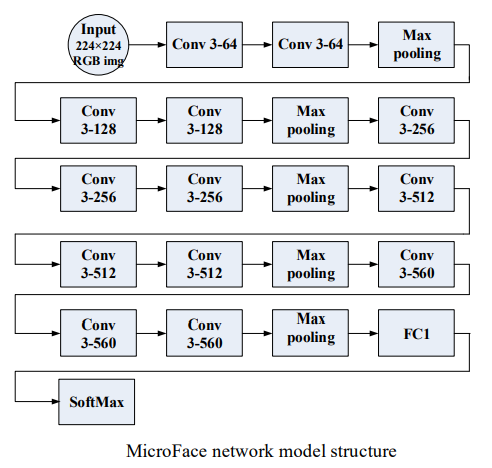
The number of convolution kernels in module 5 is increased from 512 to 560 to extract more abstract image features and improve the feature value extraction function and recognition rate of the VGG-16 network.

The polling layers is also modified. The pooling layer before the fully connected layer is modified, and the maximum pooling layer is changed to the Lp pooling layer, and the kernel size of the Lp pooling layer is 7 × 7. Lp pooling is a biologically inspired pool-forming process modeled on complex cells, and through the experiment proved that its effect is better than Max pooling. The formula of Lp pooling is:



Using the Lp pooling layer to replace the maximum pooling layer can maintain image feature extraction to the greatest extent possible while reducing the two fully connected layers, effectively reducing network parameters while maintaining image feature extraction ability to the greatest extent possible.

Following the implementation of the aforementioned enhancement measures, the algorithm's network parameters are greatly reduced, and the number of parameters is reduced from 138M to 17M, effectively reducing the memory space and training time required to save the network parameters. The detailed structure of the improved algorithm (MicroFace) has been shown below:



The training process of MicroFace network is: (1) the original face image training samples are input. (2)Face images are preprocessed (normalization, whitening). (3) The results are input into the MicroFace network. (4) The iterative training is processed till get the MicroFace network after parameter adjustment. The test flow of MicroFace is as follows: (1) the original face image test samples are input. (2) Face images are preprocessed. 3) The preprocessing results are input into the trained Microface network. (4) The face recognition results is generated.

The data base is for training and testing the MicroFace network is CASIA WebFace, published by the Institute of automation, Chinese Academy of Sciences in the year 2014.It contains nearly 500000 face images of 10575 people.

There were 2007 different types of qualified faces, and 2000 people were chosen from the database. 40 images were chosen as training images and 20 images as testing images. So, 2000 40 = 80,000 images are used for training, 2000 20 = 40,000 images are used for testing, and the final number of output categories after training is 2000.

Each image must be standardized and scaled to 224x224 pixels. The image is then whitened using the ZCA algorithm before being fed into the Microface network for recognition and training.

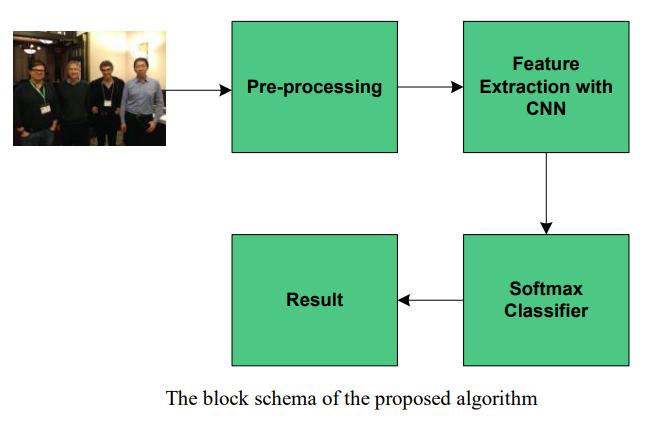
Each image must be standardized and scaled to 224x224 pixels. The image is then whitened using the ZCA algorithm before being fed into the Microface network for recognition and training. The experimental comparison results have been displayed in Table

|  |  |  |  |
| --- | --- | --- | --- |
| Network model | Number of training samples | Number of pictures/10 thousand | Recognition accuracy rate (%) |
| DeepFace | 4030 | 440 | 97.35 |
| CaffeFace | 17189 | 70 | 99.28 |
| DeepID2 | 10177 | 20 | 99.15 |
| MicroFace | 2000 | 12 | 96.26 |

The research shows that the recognition rate of the improved algorithm is 96.26%, which is enough to meet the requirements of future applications according to the researcher.

* 1. **Paper** ([1] Coşkun, 2017)

CNNs are a type of Neural Network that has shown to be extremely effective in areas such as image recognition and classification. CNNs are a type of feed-forward neural network with many layers. CNNs are made up of filters, kernels, or neurons with learnable weights, parameters, and biases. The proposed algorithm is built on the CNN architecture. The block diagram of the architecture was given bellow:



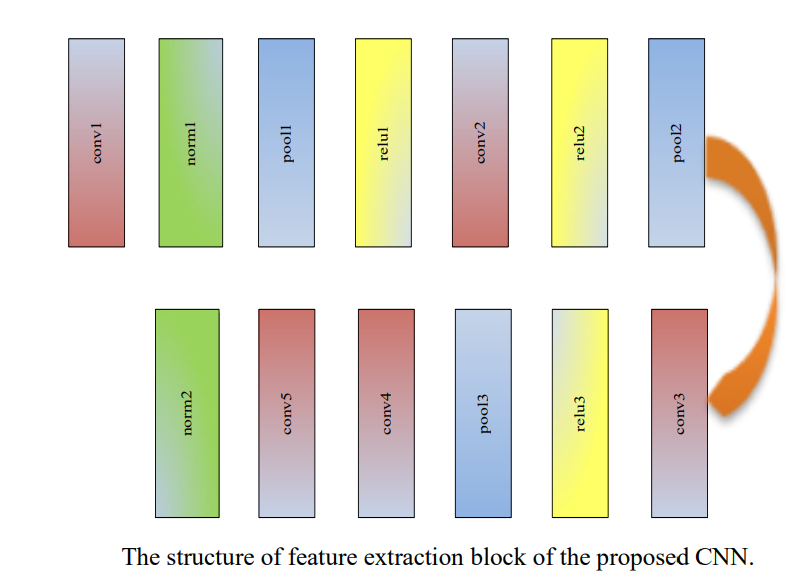
The algorithm was built in three steps as below:

1. Resize the input images as 16x16x1, 16x16x3, 32x32x1, 32x32x3, 64x64x1, and 64x64x1.

2. Construct the CNN structure with eight layers made up of convolutional, max pooling, convolutional, max pooling, convolutional, max pooling, convolutional, max pooling, convolutional, max pooling, convolutional, max pooling, convolutional, and convolutional layers.

3. After extracting all features, use Softmax classifier for classification.

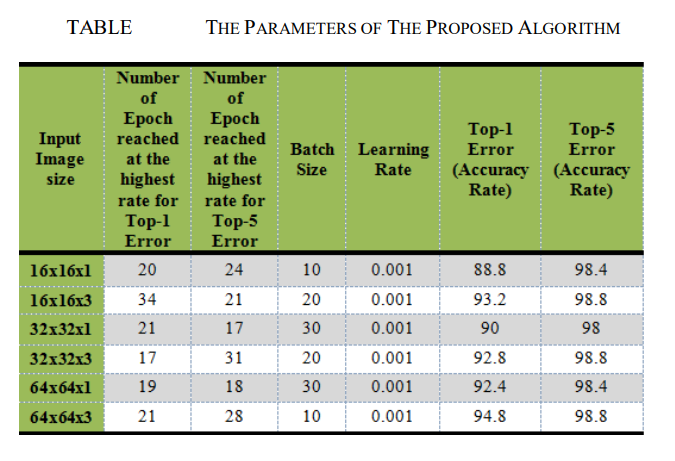
The structure of feature extraction block of the proposed CNN is illustrated bellow:



To train and test the network Georgia Tech face database was used. The database contains images of 50 individuals taken in two or three sessions between 06/01/99 and 11/15/99 at different times at the Centre for Signal and Image Processing at Georgia Institute of Technology. Each individual in the database is represented by 15 colored JPEG images with cluttered background taken at resolution 640×480 pixels. The average size of the faces in these images is 150×150 pixels. The images show frontal and/or tilted face with different facial expressions, lighting conditions and scale. All of the face regions in the images in the database were resized as 82×94.

To design the architecture Beta23 version of MatConvNet software tool was used. After pre-processing stage, size of each image was changed as 16x16x1, 16x16x3, 32x32x1, 32x32x3, 64x64x1, and 64x64x3. 66% of images were assigned as training set, 34% as test set. The architecture was trained for 35 epochs.

CNN was evaluated according to top-1 and top-5 errors. Top-1 error rate checks if the top class is the same as the target label and top-5 error rate checks if the target label is one of your top five predictions.

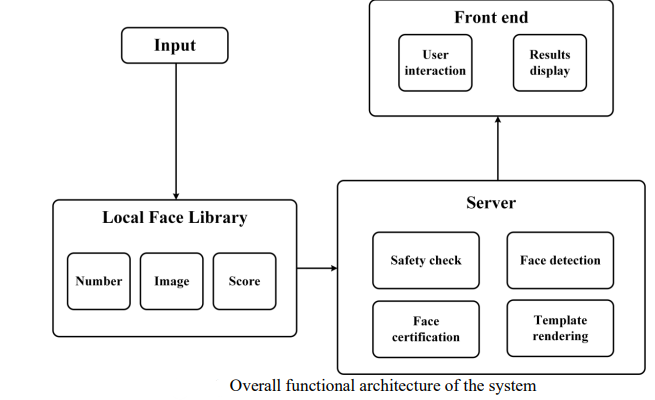


From the table it was seen the lowest top-1 error rate was obtained from 64x64x3 sized image.

* 1. **Paper** ([1] Xu, 2020)

The system was created using software engineering theory. The purpose of the face recognition management system designed in this paper is to complete the student identity authentication by using the unique biological information of the face.

Two recognition technologies are included in the field of face recognition technology based on the purpose that the system must achieve. The first is a 1: 1 face authentication technology, which requires the system to determine whether any two given images belong to the same image. The essence of face recognition in this mode is a binary classification problem. There are only two outcomes: yes or no; the second is a 1: N face recognition technology: when a face image is provided, the face closest to the face must be retrieved from the database to learn their identity. According to the analysis in this section, combined with the practical application of the two types of identification technology, the system can be divided into functional architectures as shown in Figure:

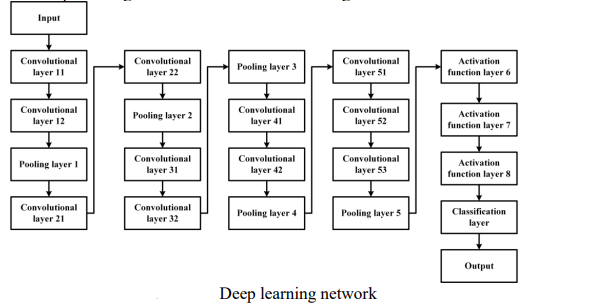


The system is deployed using a Client-Server (C/S) architecture, as shown in Figure. The system's front end includes user interaction and result display modules. In the user interaction module, the user interacts with the system and performs tasks such as uploading face images and querying face authentication results. The face discrimination result obtained from the backend is returned to the Results display module for users to view.

The back end is divided into four functional modules: security check, face detection, and face certification, template rendering.

In the Security check module, the system uses a web crawler to expand the local face image database, and then performs security verification on new images when the database is expanded, ensuring the security of the image source. When the system uploads the user's image, it must determine whether the image transmitted by the user contains the image information of the face. When the module determines that the image contains face information, the system will perform additional image processing, such as alignment and normalization. The system's core module is face certification. This module invokes the designed deep learning algorithm to complete the 1: 1 face authentication. Furthermore, this module can be used for face search, in which the user enters the most similar results from a local database. The template rendering module is primarily configured from the standpoint of system implementation. The system employs Jinja2 of Flask in Python for the design of the user interface. When implementing the interface, Jinja2 must define the base template for hierarchical design.

In the system's face certification module, an external algorithm interface was designed shown in Figure:

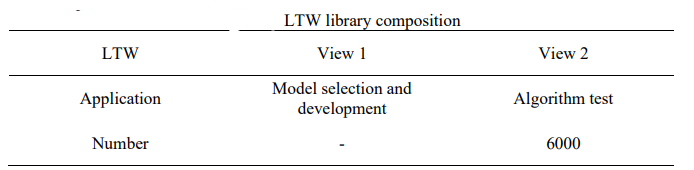


There are 11 convolutional layers, 4 pooling layers, and 3 fully connected layers in this network. The activation function layer chooses the Relu function, while the classification layer chooses the Softmax function.

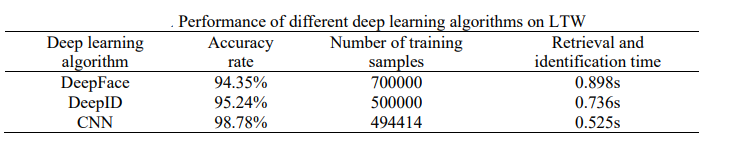
To train the network CASIA-WebFace standard face library was used. It contains 494414 face images of 10575 people. The pre-processing process of the image includes alignment and normalization of human faces and converting the image into 100×100×1 grayscale image.

The training process includes two processes, forward propagation and backward propagation. After training, multiple training models can be obtained and the network with the highest accuracy is taken as the final face recognition model.

When measuring network performance, this article uses the LTW standard face database. In this library, 13233 face images of 5749 people are included. According to the LTW test standard, the composition of the database is shown in Table below:



The test results of this paper are compared with existing deep networks shown in Table:



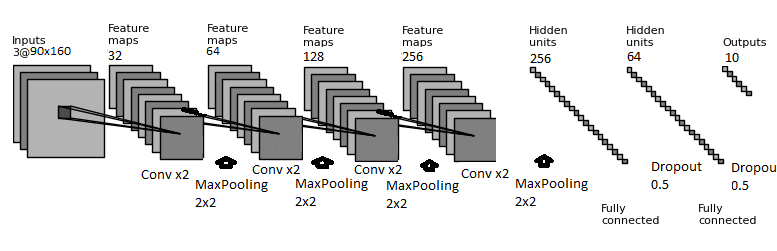
The table shows that the accuracy rate of the face recognition method proposed in this paper is higher than that of other deep networks, with a rate of 98.78 percent. At the same time, the retrieval and recognition time of the face image in this method is only 0.525s, which is significantly faster than other deep networks. This network can also be used as an algorithm module and be linked to any face recognition or management system.

* 1. **Paper** (Younis, 2017)

For the Face Recognition task, TensorFlow was used to work on a realistic private dataset that was gathered to match the specific project or problem at hand, which is recognizing students' faces in order to take class attendance at the university. The UJ Face database was created by photographing fifty male and female students at the University of Jordan in a variety of poses, zooming, orientation, illumination, and partial occlusion. Before being standardized to have a zero mean and unity standard deviation, the photos were scaled to 90 by 160 pixels.

The CNN architecture is used because it is more powerful than regular DNN. The AlexNet by Krizhevski et al. was used as a starting point for our network. Eight convolutional layers, four Max Pooling layers, two fully connected layers, and an output layer make up the CNN. A validation set of 20% of the database was chosen at random for cross-validation.

Two convolutional layers extract 32 5x5 filters with ReLu as the activation function for each layer, and the output of the activation function is fed to a 2x2 window max-pooling. The following layers use the same architecture, with a few tweaks to the number of filters, filter sizes, and max-pooling layer specifications. The figure depicts the exact architecture. Then, with 256 neurons and 64 neurons, two fully connected layers are added, each receiving the flattened output of the previous layer.



The Early Stopping method was used to reduce overfitting. If the loss function is not reduced for a certain number of epochs, the model stops learning new weights. The learner's performance on data outside of the training set improves as a result of this. Finally, a ten-neuron layer with a sigmoid activation function is added to the conventional neural network's final output.

To measure the distance between the model output and the true output from the training dataset, categorical cross-entropy was chosen as the loss function, which is an appropriate cost function for multi-class classification problems. The "RMSprop optimizer" will be used to reduce the loss value by normalizing the gradients based on the magnitude of recent gradients. The maximum number of epochs was set at 100, and the batch size was 64.

The results of the Face Recognition model validation accuracy are shown in the graph; validation accuracy began at 22% and quickly increased to over 80% after 10 epochs. After 40 epochs, it reached 98.5 percent correct classification for the test images, and after 80 iterations, it reached 100 percent correct classification for the test images. Because the loss did not change for four consecutive epochs, the model met the Early Stopping criterion and stopped at epoch 88 rather than 100.

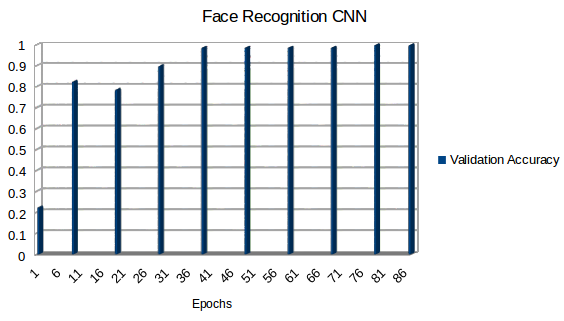
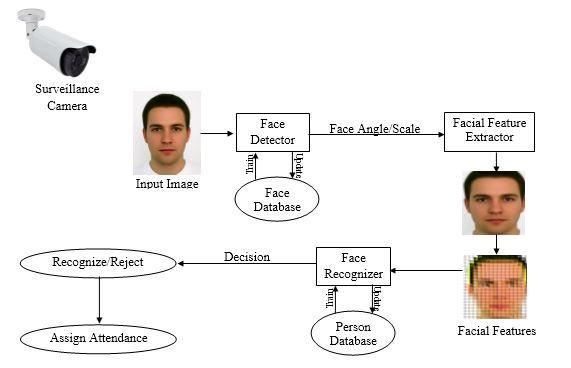


Fig: Faces Recognition CNN Validation accuracy

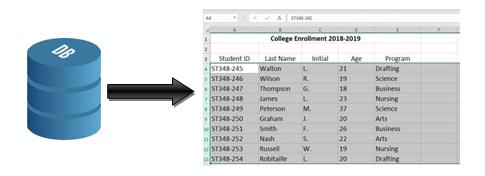
* 1. **Paper** (Nandhini, 2019)

The goal of the proposed system is to record each student's face and store it in a database for attendance purposes. The student's face, as well as the student's seating and posture, must be captured in such a way that all of the features of the student's face can be detected. The system takes a video, the face is recognized, and the attendance information is updated as a result of additional processing.

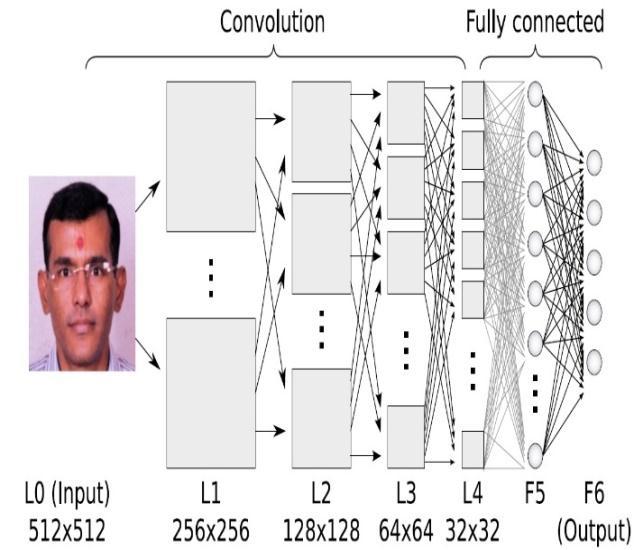
The basic operating premise of the project is that video data is converted into a picture in order to detect and recognize it. Furthermore, attendance includes a recognized image of the student; otherwise, the student is marked absent in the database.



The camera is set at a certain distance inside a classroom to capture movies of the entire class's frontal views. The acquired video should be converted into frames per second for faster identification and recognition of the students' faces. The students' names are entered into an excel spreadsheet as part of the post-processing procedure. An excel sheet can be used to track the students' attendance on a weekly or monthly basis.



A CNN (Convolution Neural Network) employs a system similar to a multilayer perceptron that is designed to process data more quickly. An input layer, an output layer, and a hidden layer with multiple convolution layers, pooling layers, fully connected layers, and normalization layers make up the CNN layer. The reduction of limitations and improvements in image processing efficiency results in a system that is far more effective and easier to train for image processing and natural language processing.



* 1. **Paper** (K. Yan, 2017)

Three convolution layers, two pooling layers, two full-connected layers, and one Softmax regression layer make up this network. The feature extractor and classifier, which can extract and classify facial features automatically, are trained using the stochastic gradient descent algorithm. The over-fitting problem is solved using the Dropout method. During the training and testing process, the Convolution Architecture for Feature Extraction framework (Caffe) is used. The ORL face database and the AR face database based on this network have a face recognition rate of 99.82 percent and 99.78 percent, respectively.

**The structure of the nine-layer network**

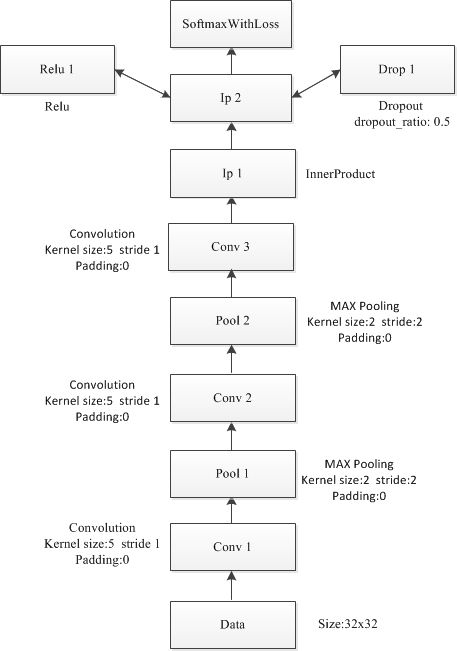


Fig-1

Each connection layer denotes a linear mapping of various data types. The figure depicts the network. Multiple feature maps make up the convolution and pool layers, and each feature map is made up of multiple neurons. Each layer's feature map is the next layer's input, and the convolution layer's feature map can be linked to some of the previous layer's feature maps.

Each convolution layer's input is the output of the upper layer, convoluted by several convolution kernels, just like a traditional neural network. The convolution kernels are applied repeatedly in each sensory field across the entire region, and the result is a feature map of the input image. The convolution kernels, which include the weight matrix w and the bias b, are the contents that the convolution layer will learn. The size of the convolution kernel in this paper is 5 x 5. The "Xavier" algorithm is used to initialize w, and 0 is used to initialize b. They will be determined by the network training process in the end. The layer's mathematical expression is:

The layer is represented by l, the convolution kernel is represented by k, the bias is represented by b, and the feature map is represented by M\_j.

**Pooling Layer:**

The dimension of the output feature maps obtained after the convolution layer calculation is usually not greatly reduced. If the dimension remains constant, a large amount of computation will be required, making the network learning process more difficult and less likely to produce a reasonable result. The pooling layer is a non-linear down-sampling method that reduces the dimension of the feature map. Each feature map that has been put into the pooling layer is sampled in the network, and the number of output feature maps remains the same, but the size of each feature map is reduced.

**Full-connected layer:**

Following a continuous stack of convolution and pooling layers, the network will typically have a number of full-connected layers near the output layer. These fully-connected layers combine to form a multi-layer perceptron (MLP), which acts as a classifier. Two fully-connected layers are used in this paper, each of which is connected to every neuron in the previous layer. The layer's mathematical expression is:

Where n is the number of neurons in the preceding layer (l-1), wlji is the weight for connection from neuron i in layer (l- 1) l to neuron j in layer (l) and blj is the bias of neuron j in layer (l), and f represents the activation function of layer(l).

**Softmax regression layer**

At the network's final layer, the softmax classifier is used, which has a strong non-linear classification ability. The Softmax classifier is a multi-classifier that can perform not only the dichotomy but also multiples (greater than 2). Assume that m samples can be classified into k classes.

training set is , , the Softmax regression function is:

(3)

where , and it presents the probability that belonging to class is the parameters of the model.

The loss function of the model is:

is indicative function, if is true, then , otherwise

**Training the network:**

In this paper, the ORL and AR face databases were trained and tested using the Caffe framework. ORL's face database contains 40 people, each with ten photos, for a total of 400 photos that include facial changes, small posture changes, and scale changes of less than 20%. The AR face database contains 100 people's faces, 50 men and 50 women, each with 26 photos, for a total of 2600 photos. Face expressions, block changes, light changes, and other changes are all present in the images. The faces of two people are chosen at random from the ORL face database, as shown in Fig. 2. As shown in Fig. 3, one person's picture is chosen at random from the AR face database.

Fig. 2: ORL database sample Fig. 3: AR database sample

**Data preprocessing**

To begin, combine the images from the two databases as follows: All of the images were converted to 32x32 pixel size using the Caffe tool, and then mirror symmetry was applied. Then, from [0–255] to [0–1], normalize the input image data. Finally, the training set was chosen from 90% of the human faces in the database, while the test set was chosen from the remaining 10%.

**Network training algorithm** the stochastic gradient descent method which has the fast convergence rate is used. The learning rate (base\_lr) is initialized to 0.01. During the training process, the learning rate is updated as:

Base\_Ir\*(1+gamma\*iter)^(-power) (5)

In this paper, iter =100, gamma = 0.0001, power = 0.75, where iter =100, gamma = 0.0001. At the same time, each iteration traverses all of the training set's batch blocks, updating the network parameters after traversing one batch block, and updating the network parameters once a batch is completed. The formula for updating is as follows:

Where is the iteration number, presents partial derivative of the loss function in the group, is the impulse parameter, is the learning rate which equals to base . Attenuation coefficient in is recommend as 0.0005. Although is small, experimental results show that it can effectively improve the classification accuracy.

The loss value and accuracy when training the ORL face database are shown in Figures 6 and 7. These figures can be easily obtained after training by using the Caffe tools to analyze and process the training data.

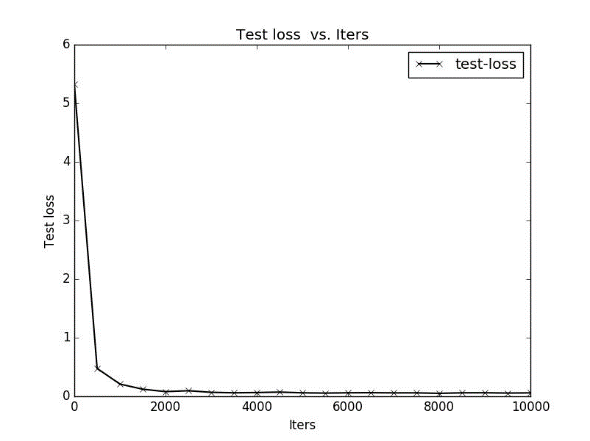
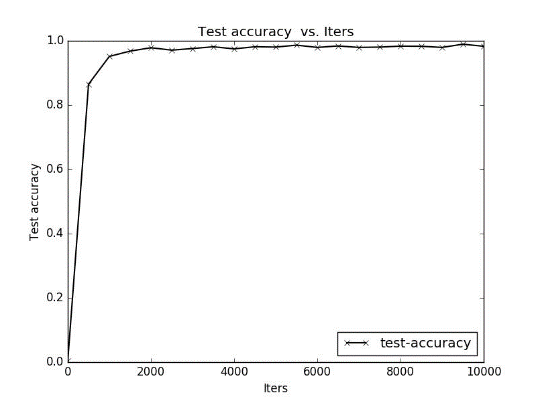
 

Fig. 4: The change of loss when training the AR database Fig. 5: The change of accuracy when training the AR database

The network has good convergence, as shown in Figures 4 and 6, and the loss value can be reduced quickly. After 2000 iterations, it reaches a stable state, which is close to 0. As shown in Figures 5 and 7, the network model can quickly achieve a 90% accuracy rate, and is close to 97 percent after 2000 iterations. It rises slowly and steadily as the number of training iterations increases, eventually stabilizing at around 98.8%.

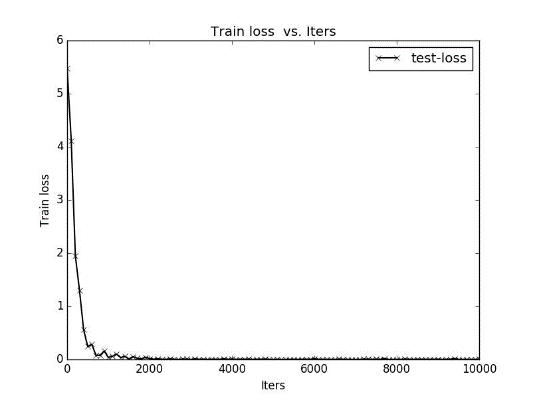
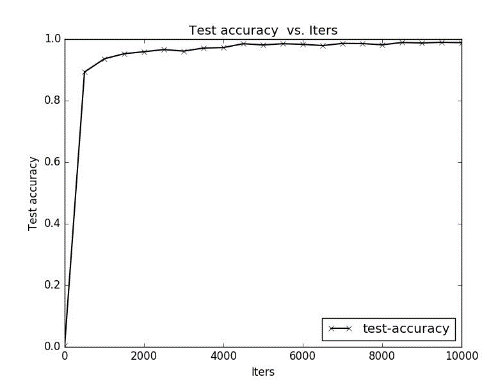
 

Fig. 6: The change of loss when training the ORL database Fig. 7: The change of accuracy when training the ORL database

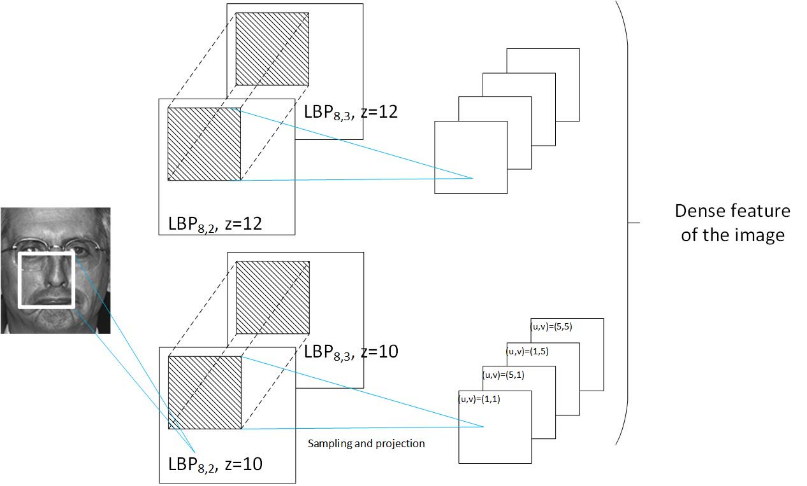
As a result, not only does the network model proposed in this paper have good convergence, but it also has high accuracy. The AR face database has a 99.78 percent correct rate of face recognition, while the ORL face database has a 99.82 percent correct rate. Furthermore, the network is resistant to changes in facial expression, regardless of whether or not there is occlusion.

* 1. **Paper** (Xi, 2016)

Face recognition is made possible by LBPNet, a powerful deep learning architecture. LBPNet's architecture is split into two parts: a deep network for feature extraction and a regular network for classification. In the deep network part, two layers are hierarchically connected to extract high-level over-complete representations of the images, using LBP and PCA filters, respectively. Two of these networks are linked to the classification networks, allowing two images to be used as input. As a result, the extracted features can be used to calculate similarity. The following two subsections go over the specifics of each layer.

**LBP Filter Layer:**

The LBP operator, LBPP,Ru2, Lables each pixel gc  in the the image by thresholding its P surrounding points



gp (p ∈ [1, P]) and then stacking labels lp (defined in Eq. 1) into one binary string

(1)

A circle with a radius of R and a center at gc is used to sample the points. Also, the labels are encoded using the unique pattern (denoted by u2 in the operator). The filter's feature is formulated as follows:

(2)

where z is the filter size and B(v) is 1 if v is true and 0 if v is false. It's worth noting that the square root of the LBP histograms is used to boost discrimination ability here. The image can be turned into a 3-dimensional feature cube by replicating the kernel. The computation is repeated in the LBP filter layer using multiple kernels subjected to different combinations of LBP radius, r, and filter size, z, to capture multi-scale representations of the image. The image's multi-scale LBP histogram features are represented by the features obtained in this layer. When viewed as a single feature vector, they have a high dimensionality, which in this experiment can reach over 10 million.

**PCA Filter Layer:** The PCA filter reduces the dimensionality of input features while improving their ability to discriminate. The input features are first sampled and concatenated into the vector pz, which is defined by.

(3)

where the filter size is M × M, s is the sampling stride, hr,z(u, v) is the feature vector located at the u-th column and v-th row of the feature cube generated by the LBP filter with sample radius r and size z, and (u, v) denotes the sampling start point. Because the feature extraction in the previous layer was done in a dense grid, the features are highly redundant—in general, two neighborhood features can share up to 90% of the same LBP labels. As a result, features are sampled to reduce the length of the resulting vector while maintaining critical discriminative information.

The PCA filter only computes features that are generated by LBP filters of the same size in this case. This is similar to CNN's partial convolutional layer connections. This is done to make computations easier and to improve discrimination abilities.

The dimensionality can then be reduced using PCA projection after obtaining the concatenated vector. PCA looks for a transformation matrix, W, for a given matrix A with a zero mean, as a result of which the reconstruction error is minimized, ||A − WT A||. The solution is known as the matrix constructed by the first n eigenvectors of the covariance matrix C = AT A. In our case, the input matrix, A, is formed by

(4)

where is defined as

(5)

The total number of features in the training set is given in Eqn. 5. By subtracting the mean vector of the entire training set from itself, the extracted feature (pi) yields another vector (qi). High variability of images in the context of face recognition generally corresponds to changes in illumination, facial expression, and so on. Because the discriminative information is uniformly distributed over all directions of the data, the impact of high variability can be reduced by down-weighting the high variance directions while increasing the weak ones. We then use whitening to normalize all of the eigenvectors, and the transformation matrix is expressed as

(6)

where is the eigenvalue of the corresponding eigenvector . The output feature is then extracted by

(7)

Wi is the PCA transformation matrix that has been whitened. Multi-scale representation, like the first layer, can be achieved by combining multiple filters with various parameters. The starting point (i, j) is changed here, but the other parameters remain the same. Figure 2 depicts a set of four filters with various starting points. Figure 1 depicts the first two layers of the deep part of LBPNet. Patch-based overcomplete features of the image are represented by the deep network outputs.

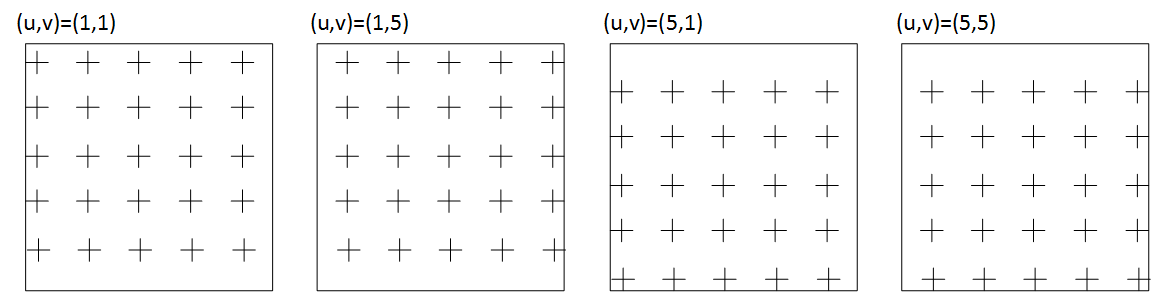


Fig. 2. 4 different PCA filters. Only vary the starting point of sampling and keep other parameters unchanged.

**Similarity Measurement**:

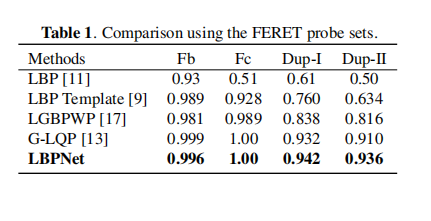
Two deep networks are linked together to take two images as input. The upper layer's extracted features are divided into two subsets, one for each image. The regional similarity scores, δi, are computed pairwise between two corresponding features. Cosine similarity, an angle-based measure, is used here.

(8)

where σi, σi are two features from upper layers, respectively. The regional similarity of two faces in a specific scale is represented by the output of these layers. The similarity of different pairs of features is assumed to contribute equally to the final score. After that, the result is calculated by averaging all regional similarities.

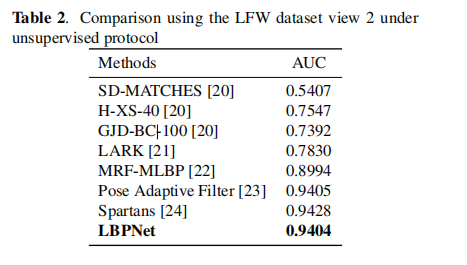
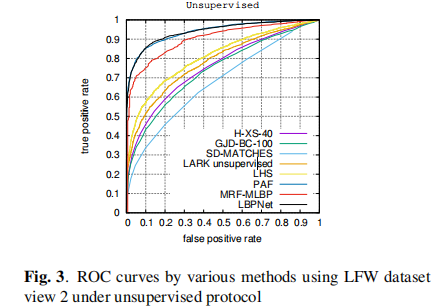
**Experiment on Face Identifification: FERET**

The well-known Face Recognition Technology (FERET) dataset was used to test LBPNet's ability to identify faces. This dataset contains 1, 196 controlled images sorted into one gallery set and four probe sets: (i) Fb set, taken in the same condition but with different facial expressions, (ii) Fc set, taken in different light conditions, (iii) Dup-I set, taken one minute to 1031 days after the gallery set, and (iv) Dup-II set, a subset of Dup-I taken at least 18 months later. The network was trained for four hours on a Linux node with 16 CPU cores and 32GB of RAM, and the classification took less than a minute. The image's center region was cropped to a size of 150 x 130 to remove the majority of the background. The images were also preprocessed according to Tan et alrecommendations. .'s LBP operators from three different companies, {LBP u2 2,8, LBP u2 3,8, LBP u2 4,8}, and two different filter sizes, z = {11, 12}, were used in the first layer. For the second layer, the filter size was 110, sampling stride in the filter and stride of the filter were s1 = z, s2 = 10, respectively, the starting point of sampling was at i , j ∈ {1, z/2} to uniformly sample all features in the filter region, and the first 2000 dimensions of PCA were selected to retain most of the discrimination information. As the results of the LBPNet and other stat-of-the-art methods shown in Table 1, LBPNet achieved comparable (in Fb), same good (in Fc) or better (in Dup-I and Dup-II) results.



**Experiment on Face Verifification: LFW:**

The proposed framework for face verification was evaluated using the de-facto evaluation benchmark, the Labeled Faces in the Wild dataset. LFW contains 13, 233 images of the faces of 5, 749 people taken in natural light. The dataset's View 2 was chosen, which contains 6000 pairs of faces, and the evaluated system confirms or rejects the identity of two faces. As specified in, the standard 10-fold cross validation was used. The LFW experiment was carried out under an unsupervised protocol. Without any prior label information or external data, the model was trained. The LFW-a version of the dataset was used, in which commercial software was used to align the images. The centre area of 170×100 of image was cropped. Three different LBP operators, {LBP u2 1,8, LBP u2 2,8, LBP u2 3,8}, and seven different filter sizes, z = {10, 12, 14, 16, 18, 20, 22} were used in the LBP layer; the filter size in the PCA filter layer was set to 80, and the first 500 PCA dimensions were chosen; the rest of the parameters were left unchanged from the FEREST experiment.

The training took 200 hours and the classification took 10 minutes, respectively. Table 2 shows a comparison of our results to other baselines and state-of-the-art results, as well as the corresponding ROC curves in Figure 3. LBPNet's AUC is 0.9404, which places it third among all networks. This value, on the other hand, is very close to the first (0.9428) and second (0.9405) bests.

Local Binary Pattern Network (LBPNet), a novel tool for face recognition, was proposed in this paper. The successful LBP descriptor and the deep learning architecture inspired the key ideas in LBPNet. By replacing its convolutional kernels with off-the-shelf computer vision descriptors, LBPNet maintains the topology of CNN while avoiding the costly model learning on massive data. To evaluate the proposed LBPNet, extensive experiments were conducted using two public benchmarks (FERET and LFW). When compared to other state-of-the-art methods, the results showed that LBPNet performed well in these benchmarks.

* 1. **Paper** (Kakarla, 2020)

TensorFlow, an open-source Python library, is used to create the proposed Convolution Neural Network (CNN).

The main components of the application are:

1) Front-End Interface (Camera)

2) Server

3) CNN Model

4) Database

5) End Result (Display Screen)

The Front-End Interface (FEI) is a web page that takes the video frame as input from the camera and processes it using the OpenCV library. The application's back-end is the server, which connects the front-end interface, model, database, and display screen. The server receives a video frame from the front-end interface, recognizes the face, and sends it to the model, which predicts the label. The server looks to see if the attendance has already been posted. If not, attendance is recorded in the database, along with a timestamp. Face data is collected from students using an automated system that uses the system camera to capture video frames and then converts them into a dataset using the following operations:

* Identify the location of face in the video frame.
* Extract the face image and convert it into gray scale image.
* Attach the label w.r.t to the class of the image and write onto a csv file.

To achieve higher accuracy, a large amount of training data is needed. As a result, data augmentation is used to generate new samples by modifying existing data. In the latest work, the following operations are used to generate random synthetic images.

* Zoom
* Shear
* Height shift
* Rotation
* Width shift

The dataset was built using 10,029 face photos divided into four classes as the initial samples. Each class has 2,500 face photos, with the exception of class-3, which has 2,529 photos. The dataset is normalized during training to speed up processing, and the train data is supplemented during the CNN learning process. The train and test data are detailed in the table.

|  |  |  |
| --- | --- | --- |
|  | Percentage | Sample |
| Train | 85 | 8524 |
| Test | 15 | 1505 |

The proposed CNN architecture consists of 20 layers which includes:

* Two-Dimensional Convolutional Layer (Conv2D)
* Batch Normalization Layer
* Max Pooling Layer
* Dense Layer

The batch normalization layer is used to normalize the input and overcome the problem of vanishing gradient and exploding gradient, while the two-dimensional convolutional layer extracts feature from the previous input. The dropout layer is used to avoid the overfitting problem and the max-pooling layer is used to reduce the dimensionality of the input. The architecture uses a grayscale image with the shape (100, 100, 1) as input and predicts the image's class label. CNN has a total of 7,658,629 parameters, 7,656,197 of which are trainable and 2,432 of which are not.

After training, the intermediate layers of the CNN model's convolutional layers Conv2D 1, Conv2D 2, Conv2D 3, Conv2D 4, Conv2D 5 are visualized with respect to an input image.

The experiments are performed on the system with the following configuration.

* Graphical Processing Unit (GPU) used is 1X Tesla K80 with 2496 CUDA cores, 12GB GDDR5 VRAM
* Central Processing Unit (CPU) used if 1X single core hyper threaded Xeon Processors, 45MB Cache with 12.6 GB RAM and 320GB disk.

The proposed Convolutional Neural Network (CNN) is trained over 50 epochs with a learning rate of 0.001, a loss metric of category cross-entropy, and a batch size of 256 using the RMSprop optimizer. During CNN training, the ReduceLROnPlateau, EarlyStopping, TensorBoard, and ModelCheckpoint callbacks are used to reduce the learning rate if the CNN learning process isn't improving, to stop the training process if the accuracy isn't improving, to visualize model graphs, and to checkpoint the model weights for future use. Graph figures show the accuracy and loss of the CNN during training as a function of the number of epochs.

The Kappa statistic, which is used to determine inter-rater reliability, is one of the classification performance measures. Precision is the proportion of correctly predicted positive observations to total predicted positive observations, while Recall is the proportion of correctly predicted positive observations to all observations in the actual class.

Finally, the CNN model has an accuracy of 99.86 percent and a loss of 0.0057 percent, respectively.

* 1. **Paper** (S. Sharma, 2016)

A face recognition system is a computer-based program that uses pre-trained characteristics from a digital image of a person to identify or verify that person, resulting in the establishment of a face database. Face recognition has progressed into a science in recent years, with more complicated algorithms and mathematical representations for feature matching. Despite the fact that it has attracted a lot of interest from academics and commercial developers, it remains challenging to apply in real-time applications. Face recognition technologies are becoming the most powerful biometric authentication module because they are non-intrusive and non-invasive.

Face detection, face alignment, face cropping, and feature extraction are the four basic procedures that make up the FAREC system. Face alignment and feature extraction are critical tasks in face recognition systems; we used Dlib for facial alignment and Convolutional Neural Network for feature extraction in this research.

FAREC systems employ Dlib to locate and align face landmarks after receiving digital photos of the person to be recognized as input. To extract attributes, just face photographs were formerly utilized.

There are many System Models:

1. The first stage in automatic face recognition is face detection, which is a type of object detection. Face detection in digital photographs or videos is a computer vision approach for recognizing a person's frontal faces.
2. Dlib is a C library for building machine learning applications that is open source. The Dlib, which is represented by Basic Linear Algebra Subprograms, is built on the foundation of linear algebra.
3. Dlib may use BLAS to do any transformations on any expression, or it can do it directly in BLAS code.
4. Dlib works with pictures, column vectors, and any other type of structured data. Dlib allows for direct processing on any object, allowing custom kernels to be used instead of fixed length vectors.
5. Face cropping is a technique for teaching a neural network only the features of facial images.

Deep learning neural networks are the most effective way for object identification, pattern recognition, and facial recognition. The convolutional neural network is one of the most often used deep neural networks, because to its capacity to adapt to picture translation invariance. It uses three main methods to extract characteristics from face images: local receptive fields, shared weights, and pooling.

A convolutional neural network's hidden neurons layer links small, localized portions of the input image pixels. Each pixel's receptive field travels through it, making touch with each concealed layer. The buried neuron is given a bias and each connection is given a weight.

Local receptive fields connected to the hidden layer with the same stride length would all have the same weight and biases. The neurons in the first hidden layer detect the same feature in several locations. The common weights and bias are referred to as kernel or filter. Shared bias refers to the bias intended for feature maps, whereas feature map refers to the map created from the input image's hidden layer.

Face recognition methods employ maximum pooling. The convolution layer, which is used to simplify the information in the output, comes after the pooling layer. For convolutional neural networks, there are a variety of pooling approaches available, however max pooling was chosen as the optimum method in this study. The maximal activation in the 2 x 2 input areas is returned via max pooling.

This paper follows the convolutional neural network architecture combining all the techniques described above.

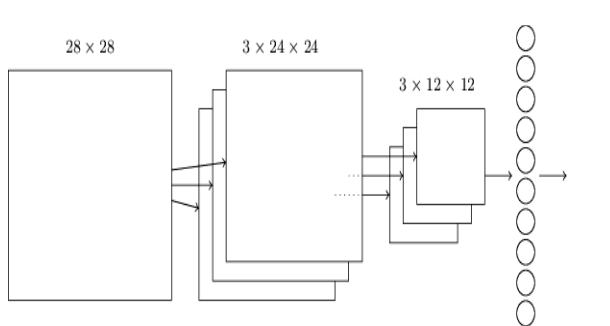


Fig. Architecture of CNN

There are three sections to the distribution of the face recognition challenge dataset. Face Recognition Grand FRGC dataset has been utilized in this study. It is made up of 50,000 digital photos organized into two sections: training and validation, and includes both controlled and uncontrolled images. To investigate the effects of light exposure on human brains, FAREC employed a controlled picture collection with 286 individual labels of 11,284 photos.

* 1. **Paper** (K. Bong, 2018)

Low-power convolutional neural network. A method based on face recognition is presented for user identification in smart devices. It has been proposed that analog–digital Hybrid Haar-like FD can improve the energy efficiency of always-on FD by 39%. Furthermore, the separable filter approximation is employed to reduce CNN burden, and transpose-read SRAM with a 7T SRAM cell is given to reduce the data read operation's activity factor.

Face recognition with always-on cameras is becoming increasingly popular. FR with always-on capability is a tempting solution for user authentication since it can provide a non-intrusive unlock for a variety of devices, as seen in Fig. Because wearable devices have limited battery capacity due to their small form factor, the FR system's always-on operation should require extremely little power while maintaining outstanding accuracy. The FR accelerator was proposed, however its accuracy was restricted due to its handcrafted feature-based algorithm. Furthermore, the digital accelerator chip failed to achieve low power consumption in the always-on FR because the whole image data generated by the image sensor must be transferred to the digital accelerator in order to detect whether or not there is a face.

Transpose-read SRAM is also available to help the CNNP use less power by permitting efficient SRAM access when using the SFA. The third section digs into the specifics of key building blocks such an analog–digital hybrid Haar-like FD and a CNN processor PE.

There are over all system Architecture:

1. Overall Algorithm Flow:

The Viola–Jones algorithm is used for the FD, and a CNN is utilized for the FV. Each stage contains a number of Haar-like filters, which are computed to determine whether the sub-window is passed or rejected at that stage.

1. Overall System Architecture:

A functional CIS for the FD and a CNNP for the FV make up the system. The 320 240 pixel array, read-out circuits, analog Haar-like filtering circuits (AHFC) with 80 20 analog memory, and an integrated image unit make up the functional CIS.

DETAILED BUILDING BLOCKS :

1. Analog–Digital Hybrid Haar-Like Face Detector:

To process the DHFU's standard Haar-like filter, the Viola–Jones algorithm is utilized. After the image sensor's rolling shutter operation, a row of the pixel array is read out, and the voltage intensities of the reduced picture are recorded in analog memory. While processing the scale with the reduction factor 4, the complete range of 20 80 analog memory is valid. 60 sub-windows are inspected each time one row of memory is updated. The 60 sub-windows in this article are handled by combining six AHFCs over ten iterations.

In the hybrid approach, the AHFC is first utilized to filter out early-rejected windows, with only the passed sub-windows being forwarded to the DHFU for the remaining stage processing. The FD achieved a 90% true positive rate and a 0.5 % false alarm rate in simulation for the Labeled Faces in the Wild (LFW) dataset.

1. CNNP Processing Element:

Many PEs in the area can work together to process a row of more than 16 words. The PE's convolutional unit consists of input registers, a 16-way SIMD-type MAC data path, and accumulation registers. The MAC unit with floating point weights is built using a shifter and adders rather than a multiplier. When compared to a standard MAC unit with a multiplier for two fixed-point inputs, it uses 43% less energy and takes up 21% less space.

1. Separable Filter Approximation and Transpose-Read SRAM:

Each of a CNN's multiple convolutional layers can be subjected to the SFA independently. In this piece, the CNN's total load is reduced by 2–3. The accuracy degradation for the FV in the LFW dataset was limited to less than 1%. The T-SRAM is based on a 7-transistor SRAM with a single decoupled read MOS connected to both a read bit line and a write word line. To access cells in a row in ordinary read mode, the horizontal line becomes the RDWL; however, in transpose-read mode, the vertical line becomes the RDWL.

The CNNP can analyze up to 77 faces per second, whereas the functional CIS imager needs 24–96 W. Peak power consumption with maximum PE usage is 5.3 and 211 mW, respectively, with related energy efficiency of 1.06 and 2.11 nJ/cycle. On the LFW dataset, the SFA is only applied to the second convolutional layer, resulting in a % accuracy loss and 97.4 % accuracy.

* 1. **Paper** (Teoh, 2021)

Face recognition is now extensively utilized in a range of applications, such as unlocking phones, criminal identification, and even home security systems. The two steps of a person recognition system are face detection and identification. This paper shows how to develop a deep learning face recognition system using OpenCV in Python.

In recent years, artificial intelligence has grown at a rapid speed. Human vision is used to adapt to and grasp the environments in which people live, whereas computer vision is used to see and interpret images in an electronic format. It has to be able to detect, recognize, and analyze images in the same manner that human vision does. Computer vision tries to replicate human systems.

One of the functions of computer vision is face recognition. Face recognition is a biometric method of recognizing someone based on a photograph of their face. Face recognition is the process of recognizing and verifying a person from an image or video.

Deep learning is the consequence of the development of artificial neural networks. Thomson introduced a new idea known as deep learning, which allows it to produce exceptional results in facial recognition by obtaining a decent approximation of a difficult function by adding hidden layers.

The fundamental image processing technique is explained here, with the time consumption in OpenCV and Matlab being the main focus point. OpenCV has been shown to be up to 30 times faster than Matlab, while the Erosion approach might be up to 100 times faster.

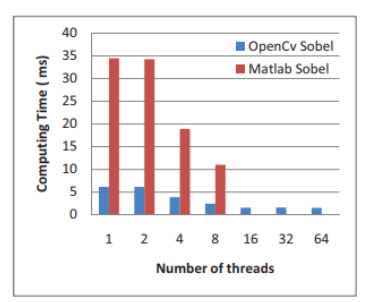


Figure 1. Time consumption between OpenCV and Matlab

Matlab is a high-level programming language based on the Java programming language, which is based on the C programming language. As a result, the computer takes a long time to interpret the code when you execute a Matlab application. On the other hand, OpenCV is mostly a C-based library. To ensure that memory allocation and leakage are not an issue, Matlab wastes system resources by overusing them. Matlab, on the other hand, is not open source and costs a lot of money to buy. As a result, OpenCV was chosen as the platform for this research.

Flowchart depicting the steps in the face recognition process. In order for facial recognition to operate, there must be an input that can be detected and verified. A fresh set of python scripts is provided to run each type of recognition. In order to conduct person recognition from a camera or picture, the python script will import the learnt classifier.

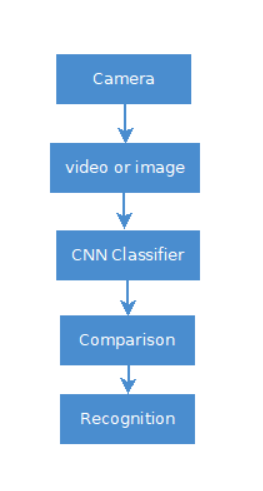


Figure . Flow chart of face recognition.

Haar feature-based cascade classifiers are employed in face recognition, and Haar Cascade is the classifier used for the frontal face. A Haar Cascade is a classifier that recognizes the source item for which it was trained. Better results are obtained by using high-resolution photographs and increasing the number of steps for which the classifier is trained.

The accuracy of the system will be tested by identifying three people several times in different locations with the purpose of evaluating how light intensity impacts the system's performance. To ensure that the information is correct, a confusion matrix is employed.

The basis for the computation is : ((TN + TP) / Total) x 100 %

Here,

TN= Genuine negative .

TP =whereas true positive .

The distance between the face and the camera has an impact on facial recognition. When the distance is around 60cm, the recommended technique has difficulty recognizing the face. However, identification happens when the distance between two persons exceeds 60cm.

The system's accuracy fluctuates depending on the amount of light. When comparing high and low lighting intensities, it is clear that higher lighting intensities result in greater accuracy. Even in low-light situations, however, the recommended method exhibits true identification.

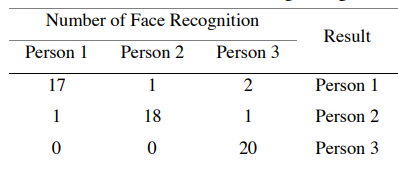


Table 1. Confusion matrix for image recognition.

Many photographs, either in a group or individually, are entered into the system to assure correctness. In those photographs, a person should have appeared 20 times. Table 1 shows the true and false detection rates of the recommended face recognition system. For the first person, 17 out of 20 recognitions are correct. Only 18 of the second person's photographs are successfully recognized, but all of the third person's photos are correctly recognized.

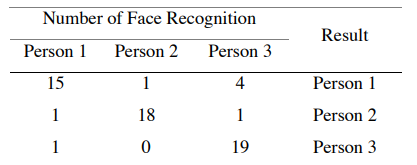


Table 2. Confusion matrix for real-time video recognition.

Person 1 has 15 valid recognitions out of a total of 20 in Table 2. When compared to Person 1, Person 3 has a decent likelihood of being recognized by the recommended system. Person 2 got 18 of the 20 questions right. The amount of accuracy reached by each participant is depicted in Figure 12.

* 1. **Paper** (Zhang Y. e., 2016)

An adaptive convolutional neural network is suggested in this study, which can decide the structure of CNN without comparing performance. The results of the face recognition experiment on the ORL face database reveal that ACNN has a superior balance between training time consumption and recognition rate.

SOM and CNN were used to develop a face recognition hybrid neural network technique. A CNN with three layers (a convolutional layer, a sampling layer, and an MLP layer) was used to identify faces. Researchers create many CNN candidates depending on their expertise and compare performance to determine the best one, which comes at a great expense and impedes CNN's advancement. Some research has been done on hardware acceleration, which is used to speed up performance comparisons, such as GPU-based programming and scalable hardware design, to solve this problem.

The traditional Conv Net design was modified by providing the classifier with both first- and second-stage features. DNN retrieves variable-scale information by dividing the maps of the final convolutional layer and the max pooling layer into several chunks with different receptive field widths or max-pooling field sizes. Although these CNN versions outperform the normal CNN structure, they still rely on prior data and experience.

In this research [19], this paper proposed an adaptable convolutional neural network (ACNN) whose structure may be defined by automated expansion based on performance needs, i.e., the scale of global expansion in ACNN is automatically decided rather than manually regulated as in ICNN. Third, unlike ICNN, ACNN manages local expansion using the recognition rate of training samples. If the global network's recognition rate of training sets is shockingly low, the network will be enlarged locally until the recognition rate reaches the target.

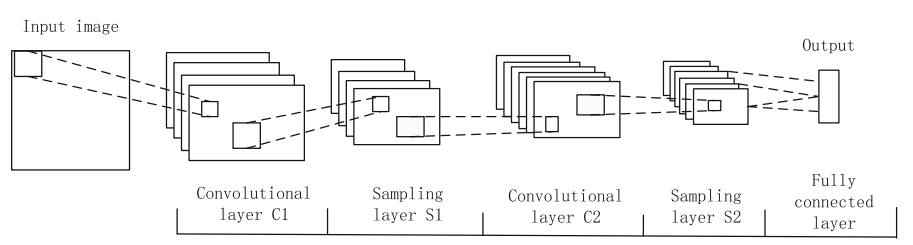


Fig. 1 Architecture of traditional CNN

Neural networks were created to classify two-dimensional pictures. The layers of a CNN are made up of a slew of 2D planes, each with a massive number of neurons. Convolution, sampling, and output are the three processes that make up a standard CNN, as shown in Figure 1.

This research presents an ACNN that can automatically build a network using a simple starting network design and a set of predetermined parameters to address some of the problems mentioned above.

The ACNN algorithm works as follows:

The network's weights are updated via back-propagation. Unlike ICNN, the initial network will be decided whether it is convergent or not after a certain number of training sessions, which is beneficial for time management. The convergent trend condition is given by the equation, where initial is the prior training's system average error, err present is the current training's average error, and T is the convergence rate threshold value of 0.1.

The global network expansion of a new branch B on the foundation of initial branch. The output of branch B is defined as:

y = f (QA + QB).

The ACNN global network has been mastered. As can be shown, ACNN's global expansion plan outperforms ICNN's in terms of minimizing human influence. The final phase is to expand the local network. The preconditions for local expansion are the same as for global expansion, and whether to develop the global or local network is artificially set.

When the system average error is less than the threshold value after global expansion but the recognition rate of training samples still falls short of the aim, a local network expansion is necessary. As illustrated, a fusion of global branches is proposed to form a local network to boost CNN accuracy..

Configuration of Face Database:

There are 40 persons in all, each with 10 different facial photographs. The first seven pictures taken by each participant are used as training samples, while the remaining shots are used as test samples. To enhance the number of training samples, we take a weighted average of three photographs randomly selected from each person's first training images to create an additional eight training photos for each participant.

The Experiment and Analysis About ACNN:

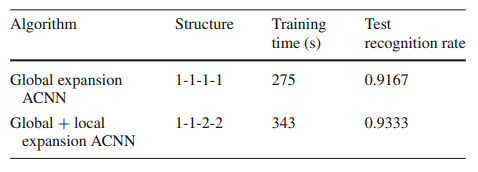


Table 1 Experiment results of ACNN

The ACNN experiment findings are listed in Table 1. Expanding the local branch based on ACNN's worldwide growth has boosted the recognition rate by 1.66 %, from 91.67 to 93.33%, proving the ACNN's practicality. ACNN drastically lowers CNN's manual interaction and develops the network automatically.

The Cross Validation Experiments:

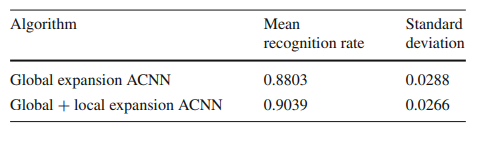


Table 2 The results about cross validation experiments

The viability of ACNN has been demonstrated by cross validation tests in this section. We chose seven pictures of each participant at random for cross validation trials and construct another eight images as training samples, with the remaining three images serving as testing samples. The experiment 1000 times, each time using a different date set. in Table 2.

* 1. **Paper** (Kamencay, 2017)

In this paper, the proposed Convolutional Neural Network (CNN) is compared to three well-known picture identification algorithms: Principal Component Analysis (PCA), Local Binary Patterns Histograms (LBPH), and K–Nearest Neigh bour (K–NN) (KNN).

Face recognition is a crucial component of biometrics, which entails comparing fundamental human features to existing data. Using efficient and effective algorithms, facial features are obtained and applied. Computers that detect and recognize faces might be utilized for a variety of purposes, including criminal identification, security, and identity verification.

Eigenfaces are what Eigenvectors are called when they're used in a computer vision problem. It integrates pixel intensity characteristics and uses Principal Component Analysis (PCA) to analyze the distribution of faces.

Local Binary Patterns (LBP) is a texture descriptor that may be used to represent faces. LBP provides a label to each pixel in an image by thresholding a 3x3 neighborhood with the center pixel value and utilizing the output as a binary.

By far the most basic machine learning/image classification algorithm is the K–Nearest Neighbor classifier. Unknown data points are classified by selecting the most common class among the K–closest instances.

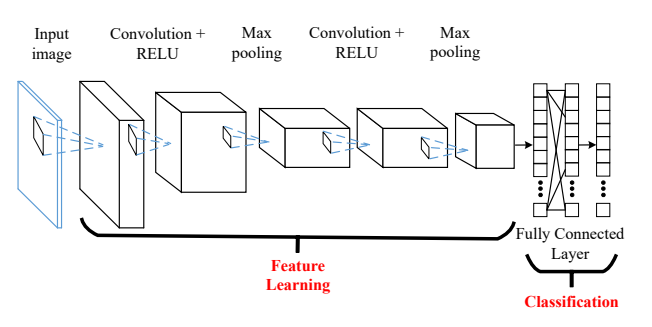
****

Fig. 1: A convolutional neural networks (CNN).

Convolutional Neural Networks (CNN) are a form of neural network that differs from traditional neural networks in several ways. They're made up of neurons that have learnt weights and biases. The whole network reflects a single differentiable scoring function, from raw picture pixels on one end to class scores on the other. The CONV/FC layers, in particular, make adjustments that are influenced by both the input volume and the parameters (the weights and biases of the neurons).

These experiments are being conducted. All of the images were aligned and normalized based on the positions of human eyes (PCA, LBPH, KNN, and suggested CNN).

Training and test photographs comprise the whole face database. The Euclidean distance was used for feature vector identification to evaluate the performance of our suggested model (accuracy of the face recognition algorithm between the test photos and all the training images).

The recommendation is CNN is a feedforward network made up of layers that transform an input picture's original pixel values to final class scores layer by layer. It may be used to categorize incoming photographs into classes based on an objective function, or it can be seen as global representations of pictures. Each face was reduced to 32x32 pixels in order to save computation time. L2regularization was used because of the limited dataset. As with block C, we used the same Max Pooling layer and Dropout value. To validate the training progress, the Soft max regression method was used.

The experiment results are divided into four categories. The proposed CNN delivered the best results for the 320 training photographs (accuracy of 98.3 % ). The KNN algorithm yielded the worst results for the 40 training pictures, with an accuracy of 71.4 %. So far, the best results have come from convolutional neural networks. It's feasible to get accuracy rates of around 98% when using complicated structures.

**Chapter 3**

# Methods

We are working on a face recognition problem in attendance, it is hard to build any completed datasets, especially in proper direction. As a result, we conducted our research using our own build dataset.

We investigated the dataset’s face recognition using learning classifier, including softmax classifier. Due to the dataset’s imbalanced in direction and environment, we used strategy to address this issue, including transfer learning and pretrained weight value from vgg16 of ImageNet Dataset.

## 3.1 Dataset Information

Our dataset consists of 430 images in 43 classes Table 3. 1. This dataset builds by taken photos of people’s frontal face from different direction in different condition randomly. We use this dataset to perform face recognition. We split the dataset in 80%, 20% ratio for train and test in each class. As a

|  |  |  |
| --- | --- | --- |
|  | Images | Class |
| Train | 344 | 43 |
| Test | 86 | 43 |
| Total | 430 | 43 |

Table 3. 1: Details of the Dataset

result, we found 344 images for train and 86 images for test from 43 classes.

## 3.2 Data Preprocessing

As we are dealing with face recognition the dataset had to be prepared in a certain way. We fixed the target size of image is 224x224 and initial batch size 16. Each class in the dataset contains 10 images. Dataset preprocessed using vgg16 models preprocessing\_input. Train images additionally rotated in range of 30 degree and zoom in range of 30%.

## 3.3 Structure of Model

In our research, we used a pretrained model. A pre-trained model is a model constructed by someone else to handle a similar problem. Instead of developing a model from scratch to handle a similar problem, use the model trained on other problem as a starting point. We found that vgg16 pretrained model trained on ImageNet dataset is best match for our classification.

**VGG16 pretrained model:** The VGG-16 architecture is a two-dimensional CNN with a 224-pixel input size Figure 3. 1. It has a total of 16 convolutional layers, with 64 filters in the first, 128 filters in the second, 256 filters in the third, and 512 filters in the fourth and fifth blocks. In each convolution, a kernel of size 33 is employed. A window with a stride of 2 is used to run all max-pooling layers. Linear rectifier units work as activation functions.



Figure 3. 1 VGG 16 Architecture

There are three fully connected (FC) layers, the sizes of each one are 4096, 1000, and 1000. FC layers was eliminated Figure 3. 2.

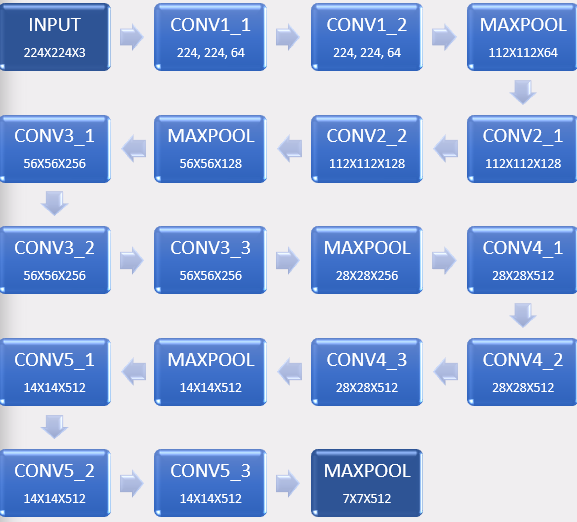


Figure 3. 2 Removing FC layers from VGG16

**ImageNet dataset:** ImageNet is a picture database arranged according to the WordNet hierarchy, with hundreds of thousands of photos representing each node of the structure. The project has made significant contributions to computer vision and deep learning research. Researchers can use the data for non-commercial purposes for free.

**Transfer learning:** Transfer learning is the process of applying a previously learnt model to a scenario. It's very popular in deep learning right now since it allows you to train deep neural networks with very little data. This is very useful in data science, because most real-world scenarios do not need millions of labelled data points to train complex models. Transfer learning occurs when a model created for one activity is applied to a different task. Fine-tuning is a method of transfer learning in which the model output is changed to meet the new goal and only the output model is trained.

The number 16 in the name VGG refers to the fact that it is 16 layers deep neural network. This means VGG16 is a pretty extensive network and has a total of around 138 million parameters. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into different categories.

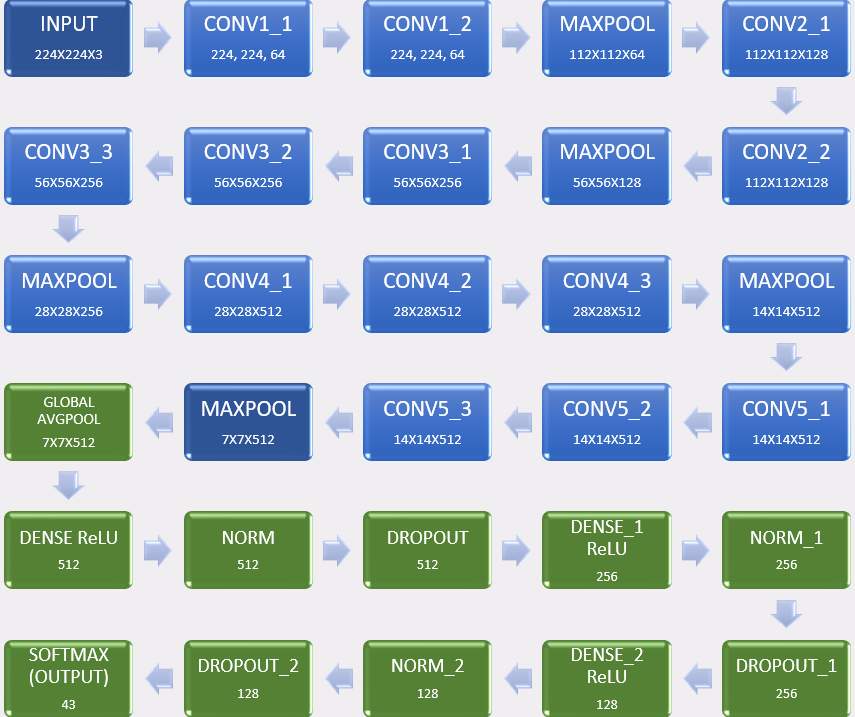


Figure 3. 3 Modified VGG16 model

Here in Figure 3. 3, added a new fully-connected layer that has an output dimension equal to the number of classes in the new data set. In new layers, takes input 7X7 and 512 input channel shape then input goes to global avgpool after that the matrices pass through 3 times to dense layer which activate ReLU activation, normalization and dropout layer with different input channel. Finally, it comes to output layer and activate softmax activation. Here in the process chart of vgg16, there is 4 fully connected layer or activation layers 3 of them are ReLU activation and last one is softmax activation. To use this model, first remove the fully connected layers and also removing the input block from our model. Because input block won’t be needed though taking input from pretrained model. Concatenated our model with vgg16 pretrained model. Now can be seen the shape of the joining point matched. That means there won’t be any error on joining.

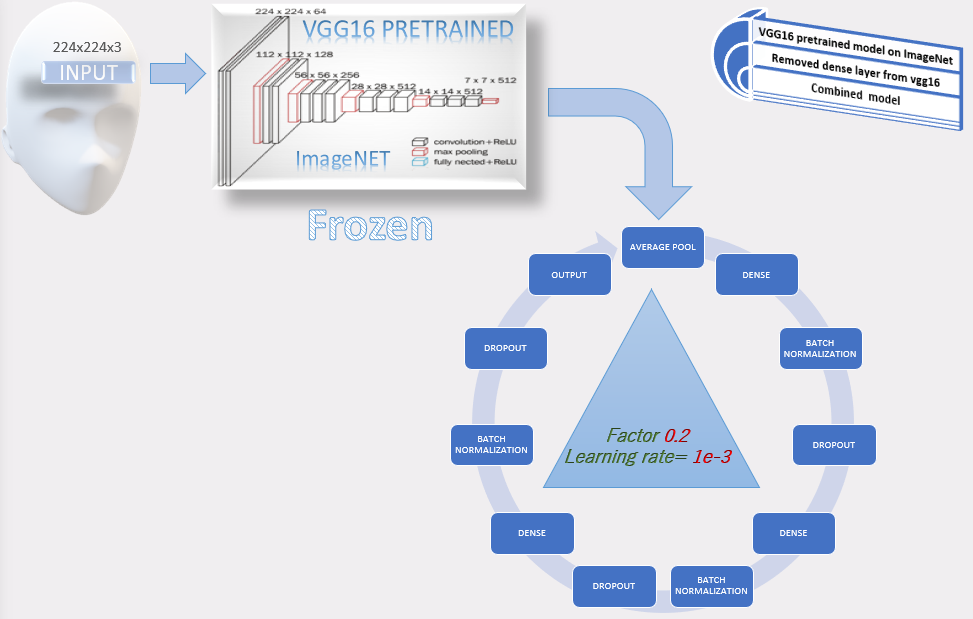


Figure 3. 4 Transfer learning workflow

Here in Figure 3. 4, can be seen it takes input first from the dataset and go through the vgg16 which already has trained value from ImageNet database. Now randomize the weights of the new fully connected layer and freeze all the weights from the pre-trained network. After that the matrices value Loop through the model. In this combined model we use transfer learning to train the network to update the weights of the new fully connected layer which learning rate starts at 1e-3, reduce learning rate by o.2 factor is specialty of this model. Here vgg16 value has been frozen or not upgrading weights value, only upgrading value on loop area. Now, Transfer learning Iterative Algorithm that starts off at a random point on the loss function and travels down its slope in steps until it reaches the lowest point (minimum) of the function. Algorithm works by Calculate what a small change in each individual weight would do to the value loss function, adjust each parameter based on it take a small step in the determined direction. Repeat steps until the value loss function is as low as possible. Reduce learning function monitor the value loss and factor by which the learning rate will be reduced.

**Chapter 4**

# Results or findings

Use To train the model, a custom dataset was created which contains 430 images of 43 people, and from that dataset, 0.8% (344 images belonging to 43 classes) images used for training and 0.2% (86 images belonging to 43 classes) images were used for validation. Model was compiled for training with Adam optimizer with learning rate of 1e-3, sparse\_categorical\_crossentropy loss function and accuracy metrics. We use ReduceLROnPlateau call back function with (monitor='val\_loss', factor=0.2, patience=2, min\_lr=0.00001, verbose=1) value. Then the model was trained up to 50 Epoch. After training we plot the accuracy and loss graph to visualize our model performance.

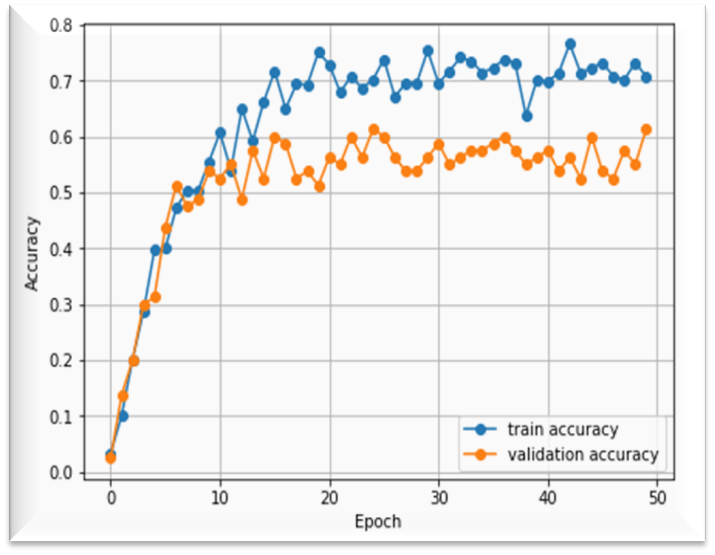


Figure 4. 1 Train and Validation Loss Graph

The graph shown in Figure 4. 1 represent the train and validation accuracy of our model. In that graph, X-axis represent number of Epoch and Y-axis represent accuracy. In the graph the line plotted with blue dot represent the accuracy during the time of training and the line plotted with orange dot represent the accuracy during validation. As we can see from the graph for first 10 Epoch the train and validation accuracy were increasing rapidly. The final training accuracy was 0.7410 and validation accuracy was 0.6125.

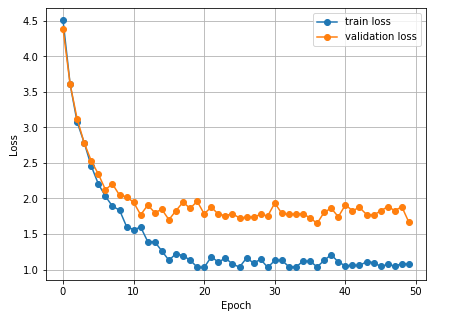


Figure 4. 2 Train and Validation Accuracy Graph

The graph shown in Figure 4. 2 represent the train and validation loss of our model. In that graph, X-axis represent number of Epoch and Y-axis represent the loss. In the graph the line plotted with blue dot represent the loss during the time of training and the line plotted with orange dot represent the loss during validation. The training and validation loss was decreasing gradually with each Epoch. The final training loss 1.0407 was 0.7410 and validation accuracy was 1.6645.

After analyzing all the data, we can see that the final validation accuracy was 61.125% which is a modest result in a fine-grained classification. Since we are using a small dataset the accuracy can be improved by using large dataset.

**Chapter 5**

# Discussion

In our research, we worked on VGG16 pretrained model using transfer learning methods. Because of time constraint, we able to develop a small dataset and trained the model with it. After the training accuracy was 74.1 percent, while the validation accuracy was slightly lower than projected at 61.13 percent. In our thesis, we have done fine-grained categorization. The purpose of fine-grained categorization is to find photos that fit into multiple subcategories within a single category. This is a difficult task due to the inherent minor variances among highly-confused groups. The accuracy would be higher if we utilized a different type of class. Another flaw was the limited data set. A dataset of 430 photographs has been used, separated into 43 categories. The model we developed will perform better if we overcome those limitations.

**Chapter 6**

# Conclusion

Attendance is taken at most institutions in a manual way, which consumes valuable time and effort. There is also some other problem with taking attendance, like giving someone else's attendance. Our goal with this research is to automate the attendance system using computer vision and deep learning using state-of-the-art technology. To do so, the transfer learning method was utilized. A pre-trained VGG-16 model was used and modified according to the needs. A dataset was created from the ground up specifically to train this network. After training, a modest result was achieved using this data set. By utilizing large datasets, there is room for improvement. If such a system can be built and used, it will be possible to avoid the hassle of taking attendance and devote that time to more important tasks. For these reasons, more research in this area is necessary.

## 6.1 Future scope

To make the system better in the future, we need to work on a module that can recognize faces from real-time video feeds and also check for eye blinks to identify real human presence so that attendance can be taken in real-time. It’s an added layer of security to prevent any unwanted attendance by showing a picture of the person. To improve the accuracy of the model, we will work on a larger dataset with more sample images of the same class to improve accuracy so that the system can be used more reliably.

# Bibliography

[1] Coşkun, M. &. (2017). Face Recognition Based on Convolutional Neural Network. *2017 International Conference on Modern Electrical and Energy Systems (MEES). IEEE*, 376-379.

[1] Nakada, M. H. (2017). AcFR: active face recognition using convolutional neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*.

[1] Pranav, K. B. (2020). Design and evaluation of a real-time face recognition system using convolutional neural networks. *Procedia Computer Science 171*, 1651-1659.

[1] Xu, S. e. (2020, May). Face Recognition Algorithm Based On Convolutional Neural Network. *Journal of Physics: Conference Series, Vol. 1550, No. 2, p. 022023*.

Heidari, M. &.-G. (2020). Using Siamese networks with transfer learning for face recognition on small-samples datasets. *International Conference on Machine Vision and Image Processing (MVIP)*, 1-4.

K. Bong, S. C.-J. (2018, January). A Low-Power Convolutional Neural Network Face Recognition Processor and a CIS Integrated With Always-on Face Detector. *in IEEE Journal of Solid-State Circuits*, 115-123. doi:10.1109/JSSC.2017.2767705

K. Yan, S. H. (2017). Face recognition based on convolution neural network. *2017 36th Chinese Control Conference (CCC)*, 2017.

Kakarla, S. e. (2020). Smart Attendance Management System Based on Face Recognition Using CNN. *2020 IEEE-HYDCON. IEEE*.

Kamencay, P. M. (2017). A new method for face recognition using convolutional neural network. *Advances in Electrical and Electronic Engineering 15*, 663-672.

Li, L. e. (2017). An incremental face recognition system based on deep learning. *2017 Fifteenth IAPR international conference on machine vision applications (MVA). IEEE*.

Lucena, O. J. (2017, July). Transfer learning using convolutional neural networks for face anti-spoofing. . *International conference image analysis and recognition*, 27-34.

Nandhini, R. D. (2019). Face recognition based attendance system. *International Journal of Engineering and Advanced Technology (IJEAT)*.

S. Sharma, K. S. (2016). FAREC — CNN based efficient face recognition technique using Dlib. *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 192-195.

Said, Y. M. (2020). Design of a Face Recognition System based on Convolutional Neural Network (CNN). *Engineering, Technology & Applied Science Research 10.3*, 5608-5612.

Teoh, K. H. (2021). Face recognition and identification using deep learning approach. *Journal of Physics: Conference Series, 1755*.

Wang, M. Z. (2017). Deep convolutional neural network applies to face recognition in small and medium databases. *2017 4th International Conference on Systems and Informatics (ICSAI). IEEE*.

Xi, M. e. (2016). Local binary pattern network: A deep learning approach for face recognition. *2016 IEEE international conference on Image processing (ICIP). IEEE*.

Yin, X. Y. (2019). Feature transfer learning for face . *IEEE/CVF conference on computer vision and pattern recognition*, 5704-5713.

Younis, K. S. (2017). A new implementation of deep neural networks for optical character recognition and face recognition. *Proceedings of the new trends in information technology*, 157-162.

Zeng, J. Z., & Wang, F. (2018). Deep convolutional neural network used in single sample per person face recognition. *Computational Intelligence and Neuroscience, 2018*, 1-11.

Zhang, H. e. (2017). A face recognition method based on LBP feature for CNN. *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE*.

Zhang, Y. e. (2016). Adaptive convolutional neural network and its application in face recognition. *Neural Processing Letters 43.2*, 389-399.

Zhiqi, [. Y. (2021). Face recognition based on improved VGGNET convolutional neural network. *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2530-2533.