National College of Engineering

(Affiliated to Tribhuvan University)

***Talchhikhel*, Lalitpur**

[Subject Code: CT755]

A MAJOR PROJECT REPORT ON

"FACIAL LANDMARK DETECTION USING CNN"

Submitted by:

Dilip Paudel	**[NCE-073-89756]**
Rajan Pandey	**[NCE-073-89765]**
Diwash Bhandari	**[NCE-073-89770]**
Sailesh Sapkota	**[NCE-073-89778]**

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

March, 2021

FACIAL LANDMARK DETECTION USING CNN

Submitted by:

Dilip Paudel	**[NCE-073-89756]**
Rajan Pandey	**[NCE-073-89765]**
Diwash Bhandari	**[NCE-073-89770]**
Sailesh Sapkota	**[NCE-073-89778]**

Supervised by:

Er. ANUP SHRESTHA

(Department of Computer and Electronic Engineering)

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

National College of Engineering

Talchhikhel, Lalitpur

March, 2021

COPYRIGHT

CERTIFICATE

NATIONAL COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING
APPROVAL LETTER

The undersigned certify that they have read and recommended to the Institute of Engineering for acceptance, a project report entitled "**FACIAL LANDMARK DETECTION USING CNN**" submitted by Dilip Paudel, Rajan Pandey, Diwash Bhandari, Sailesh Sapkota in partial fulfillment for the degree of Bachelor of Engineering in Computer Engineering.

……..	……..
Supervisor	External Examiner
Er. Anup Shrestha	Department of Computer, Electronic and Communication (IOE)
……..	……..
Er. Mohan Maharjan	Er. Pradip Adhakari
Deputy Head of Department	Head of Department
Department of Computer, Electronic and Communication Engineering (NCE)	Department of Computer, Electronic and Communication Engineering

ACKNOWLEDGEMENT

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our highly respected and esteemed guide **Er. Anup Shrestha**, supervisor of our project for his valuable guidance, encouragement and help for completing this work. His useful suggestions for this whole work and co-operative behavior are sincerely acknowledged. We would like to express our sincere thanks to **Er. Pradip Adhakari**, Head of Department of Electronic and Computer for giving us this opportunity to undertake this project. We would also like to thank **Er. Mohan Maharjan**, for whole hearted support. We are also grateful to our teachers **Er. Sarmila Bista** and **Er. Subash Pandey** for their constant support and guidance.

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

ABSTRACT

Facial landmark detection using CNN involves reading of input images from the camera and detects any faces with their landmarks for applying filters. Human beings are able to recognize faces while learning from childhood. But the same task for machines is much more complex because of complexity of the occlusion or illumination and variation in pose. With the advent of deep learning with neural network techniques, the accuracy for face detection has increased drastically. The system uses Haar-Like Features algorithm for face detection by detecting presence of facial features in the given image and CNN is implemented to detect facial landmarks by extracting features from image. The CelebA dataset was used to build a model which has 35000 face images with 136 landmarks and 0.05 validation errors was obtained with the help of absolute mean error.

Keywords: ***Facial landmark, CNN, Haar-Like Features, occlusion, pose, CelebA.***

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATION

2D: Two Dimensional

AI: Artificial Intelligence

CelebA: CelebFaces Attributes Dataset

CNN: Convolutional Neural Network

ConvNet: Convolutional Neural Network

GPU: Graphical Processing Unit

GUI: Graphical User Interface

NN: Neural Network

OpenCV: Open-Source Computer Vision

RAM: Random Access Memory

ReLu: Rectifier Linear Unit

CHAPTER 1: INTRODUCTION

1.1 Background

Facial landmark detection is a well-studied topic in the field of computer vision with many applications such as face analysis, face recognition, or face modeling; see for a review. The high variability of shapes, poses, lighting conditions, and possible occlusions makes it a particularly challenging task even today. In contrast to face recognition, where modern approaches using convolutional neural networks (CNNs) are beyond human-level performance, computers are still below par at facial landmark detection. [1]

Facial detection is the ability of a machine to receive input from multiple sources like video, photographs etc. and detect the required portion of the face. Face-detection algorithms focus on the detection of frontal human faces. Significant efforts have been paid to develop representation schemes and algorithms aiming at recognizing generic objects in images taken under different imaging conditions (e.g., viewpoint, illumination, and occlusion).

The accurate identification of landmarks within facial images is an important step in the completion of a number of higher-order computer vision tasks such as facial identification, expression analysis, age estimation, and gender classification are often built upon a facial land-marking component in their methods [2] [3]. Most landmark detectors need a bounding box around the face to either initialize the algorithm or to crop the image. Thus, before applying a landmark detector in practice a face detector is needed to localize faces in the image. In this project, these two problems are looked at separately. As we focus on practical performance of different landmark detectors, we want to test them in combination with a face detector. This way we can measure performance in actual testing conditions.

For fully-automated facial landmark detection, the system uses Haar-like feature algorithm, CNN method and large labeled data-sets. The goal of this project is to apply

Deep Convolutional Neural Network to be able to identify landmarks. The advantage of this approach is that it has robustness to pose as well as illumination.

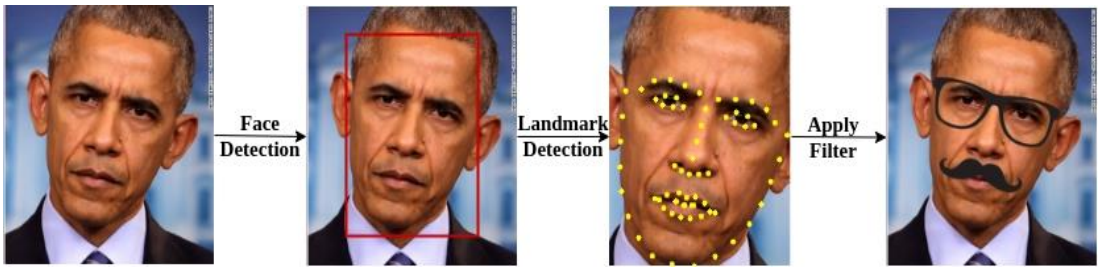


Figure 1: Applying filter after detecting facial landmark

1.1.1 Region of interest

The Regions of interest extract only those features from the data relevant for solving the defined problem. This stage detects and extracts the face from the image and discards irrelevant information such as the background. For solving this problem, a Haar-like features algorithm is used.

Haar-like feature algorithm is an appearance-based approach which is used for finding the location of the human faces in a frame or image. All human faces share some universal properties of the human face like the eye's region is darker than its neighbor pixels and nose region is brighter than eye region. This algorithm uses a combination of two main techniques: the representation of Haar-like features with an 'integral image' and a series of 'weak' classifiers boosted using the Adaboost learning algorithm.

1.1.2 Facial landmark detector

Facial landmark Detector is the system that detects key landmarks on the face and tracking them. It is done by the help of convolutional neural networks. Convolutional Neural Networks are a special type of feed-forward artificial neural network in which the

connectivity pattern between its neurons is inspired by the visual cortex. They are widely used in image and video processing such as classification, segmentation, recommendation systems and natural language processing etc. It takes a processed input image of a certain size, assigns learnable kernels to extract various features from the image and will be able to differentiate one from the other.

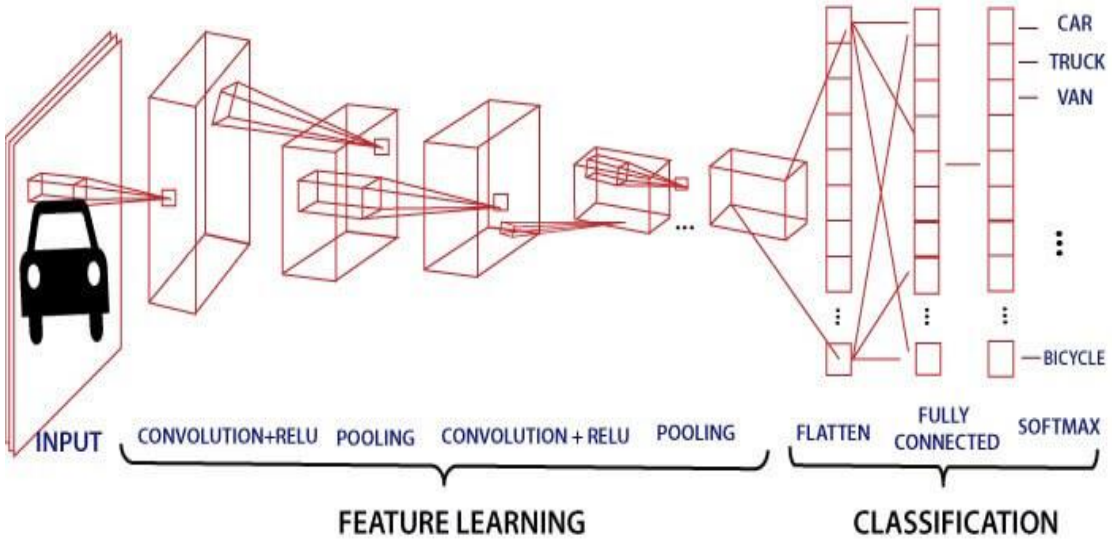


Figure 2: Model of CNN (Source: Google)

1.2 Problem Statement

- The main challenges for facial Key-point detection are that facial features have great variance between different people and different external factors but our model can accommodate with slight variation in pose.
- While algorithms like SVM were not efficient to find facial landmarks as required features from images are hand engineered, but with enough training, ConvNets have the ability to learn these filters/characteristics through the training.

1.3 Aim and Objectives

The main aim of this project is:

- To develop a system that can detect facial landmarks in the photo to apply different face filters.

The objective of this project is:

1. To use Haar-like feature algorithms for locating faces.
2. To implement CNN for detecting facial landmarks.
3. To apply a face filter in the appropriate location.

1.4 Scope of Project

Facial landmark detection is the task of detecting key landmarks on the face and tracking them. This research project includes facial landmark detection for the position of appropriate face filters in their respective place. But the landmark detection can be widely applied in many fields. Some of them are: face recognition, facial expression analysis, 3D face modeling, age estimation, gender classification etc. This project findings can help people to understand the power of CNN as it was capable of learning filters on its own and using that filter for the detection of landmarks in face with the help of appropriate data and CNN model. This project will also guide how to apply CNN for different image problems and also apply facial detection abilities in another field also like in people to analyze their facial expression to understand their behavior.

CHAPTER 2: LITERATURE REVIEW

With the advancement of technology, their application and scope has also been elongated. The researcher with their knowledge in the field of science and technology has now been in the path of finding the miracle of working of our magnificent brain by introducing AI to us and has been eager to apply the findings in the advancement of technology. In 1956, American computer scientist John McCarthy organized the Dartmouth Conference, at which the term 'Artificial Intelligence' was first adopted. The successful application of convolution networks was developed by Yenn Lecunn in the 1990s.

Several other recognition models were consulted and the major works that we followed were:

1. Facial KeyPoint's detection using Neural Network [4]:

The objective of facial keypoints detection is to find the facial keypoints in a given face, which is very challenging due to very different facial features from person to person. In this project, key point in the given image is located using deep architectures to not only obtain lower loss for the detection task but also accelerate the training and testing process for real-world applications. This paper has defined two basic neural network structures, one hidden layer neural network and convolutional neural network as the project baselines. The project objective is to locate 15 sets of facial keypoints when given a raw facial image. The dataset we use is from Kaggle's facial keypoints detection competition. There are 7049 images in total, each 4 of which is a 96 × 96 pixels image. For the experiments, evaluation metrics for regression loss, mean squared error (MSE) between the ground truth keypoint coordinate vector and the predicted one is used. For one hidden layer model, which is the simplest deep architecture among the 5 approaches, achieve the worst accuracy (loss test = 3.881). For CNN model, the loss is slightly better (3.09).

2. **Fully Automatic Facial Feature Point Detection Using Gabor Feature Based Boosted Classifiers [5]:**

In this paper the author has present a method for fully automatic detection of 20 facial feature points in images of expressionless faces using Gabor feature based boosted classifiers. The detected face region is then divided into 20 relevant regions of interest, each of which is examined further to predict the location of the facial feature points. The proposed facial feature point detection method uses individual feature patch templates to detect points in the relevant region of interest. These feature models are Gentle Boost templates built from both gray level intensities and Gabor wavelet features. The facial feature detection method was trained and tested on the Cohn-Kanade database, which consists of approximately 2000 gray-scale image sequences in nearly frontal view from over 200 subjects, male and female, being 18 to 50 years old. When tested on the Cohn-Kanade database, the method has achieved average recognition rates of 89%.

3. Deep Convolutional Network Cascade for Facial Point Detection [6]:

This paper proposes a new approach for estimation of the positions of facial keypoints with three-level carefully designed convolutional networks. At each level, the outputs of multiple networks are fused for robust and accurate estimation. Thanks to the deep structures of convolutional networks, global high-level features are extracted over the whole face region at the initialization stage, which help to locate high accuracy keypoints. There are two folds of advantage for this. First, the texture context information over the entire face is utilized to locate each keypoint. Second, since the networks are trained to predict all the keypoints simultaneously, the geometric constraints among keypoints are implicitly encoded. The method therefore can avoid local minimum caused by ambiguity and data corruption in difficult image samples due to occlusions, large pose variations, and extreme lightings. The networks at the following two levels are trained to locally refine initial predictions and their inputs are limited to small regions around the initial predictions. The dataset was created with 13, 466 face images, among which 5, 590 images are from LFW 2 and the remaining 7, 876 images are downloaded from the web.

Performance is measured with the average detection error and the failure rate of each facial point.

4. Local Evidence Aggregation for Regression Based Facial Point Detection [7]:

This paper proposes a new algorithm to detect facial points in frontal and near-frontal face images. It combines a regression-based approach with a probabilistic graphical model-based face shape model, that restricts the search to anthropomorphically consistent regions. This algorithm detects the target location by aggregating the estimates obtained from stochastically selected local appearance information into a single robust prediction. The proposed algorithm was tested on over 7,500 images from 5 databases: MMI Facial Expression database, FERET database, XM2VTS database, SEMAINE database and BioID database. Performance is measured with the interocular distance (IOD)-normalized error.

5. Random Cascaded-Regression Copse for Robust Facial Landmark Detection [8]:

In this research paper, writer presents a random cascaded-regression copse (R-CR-C) for robust facial landmark detection. Its key innovations include a new parallel cascade structure design, and an adaptive scheme for scale-invariant shape update and local feature extraction. Evaluation on two challenging benchmarks shows the superiority of the proposed algorithm to state-of-the-art methods. Moreover, the experimental results obtained on two challenging benchmarks using a sparse auto encoder demonstrate the superiority of the proposed algorithm compared to the state of the art.

CHAPTER 3: REQUIREMENT ANALYSIS

3.1 Functional Requirements

Rid	Requirements
1.	Image should be converted into grayscale.
2.	Images should be rescaled in needed format.
3.	Face should be detected in the frame.
4.	Users should be able to upload video.
5.	Users should be seeing the result for both real and non-real time data.
6.	System should be able to detect 128 landmarks in frame.
7.	Uses should be able to select face filters.
8.	The result should be displayed with a selected face filter in the appropriate position in the face.

Table 1: Functional Requirements

3.2 Non-Functional Requirement

Non-functional requirements are as follows:

Usability:

Prioritize the essential functions of the system based on usage pattern Frequently used functions should be tested for usability, as should complex and critical functions. Be sure to create a requirement for this.

Reliability:

Users must trust the system, even after using it for a long time. Your goal should be a long MTBF (mean time between failures). Create a requirement that data created in the

system will be retained for a few years without the data being changed by the system. It is a promising idea to also include requirements that make it easier to monitor system performance.

Performance:

What should system response times be, as measured from any point, under what circumstances? Are there specific peak times when the load on the system will be unusually high? The following can be the points needed to be considered in the performance of the facial landmark detection:

- It should identify landmark in the photo under few second.
- It should apply face filter instantly

Supportability:

The system needs to be cost-effective to maintain. Maintainability requirements may cover diverse levels of documentation, such as system documentation, as well as test documents.eg which test cases and test plans will accompany the system.

3.3 System Requirements

3.3.1 Hardware Requirement

- For training our deep neural network, Google Collaboratory cloud computing will be used which has following hardware requirement:
- RAM – 12 GB
- GPU - Tesla K80 GPU
- Hard-disk – 10 GB

3.3.2 Software Requirement

- It can be used on all types of system as well as all OS.
- The project is implemented in python 3.

- OpenCV for image pre-processing.
- NumPy for matrix and array operations.
- Matplotlib for plotting graphs.

3.4 Feasibility Analysis

The feasibility study determines whether the proposed system is useful and feasible to use to the organization and user.

3.4.1 Technical Feasibility

All the technical resources required for the project including hardware parts and software are easily available in the market. User must have camera to capture the image and storage device to store them and working PC to operate the software. So, there must not be a problem for us to get those things that are required for the project.

3.4.2 Economic Feasibility

The project is economically feasible and is within the range of affordable expenditure as most of the equipment and electronic devices are already available. Once the system setup is done and it starts functioning as it is supposed to.

3.4.3 Operational Feasibility

This project is quite a complex system while designing as it requires of deep learning with neural network with their different mathematical algorithm and a large dataset for it training. But once the system is implemented and starts to operate with accurate result there will operationally feasible because this software doesn’t require special training to operate in side of users. This system can be operated with basic knowledge of computer.

CHAPTER 4: METHODOLOGY

4.1 Dataset Collection

Since Convolution Neural networks is supervised Machine Learning technique, a large amount of labeled data (images) is needed for the purpose of training. A labeled data (images) is a data that has input and knows the expected output value for it. In this project, the input is the image of a person with different landmarks and a labeled landmark is normalized. In this project dataset was obtained from CelebFaces Attributes (CelebA) Dataset [9]. From the CelebA dataset repository, 35000 face images with 136 landmarks are used for training dataset and testing dataset.

Figure 3: Sample of Dataset

4.2 System Block Diagram

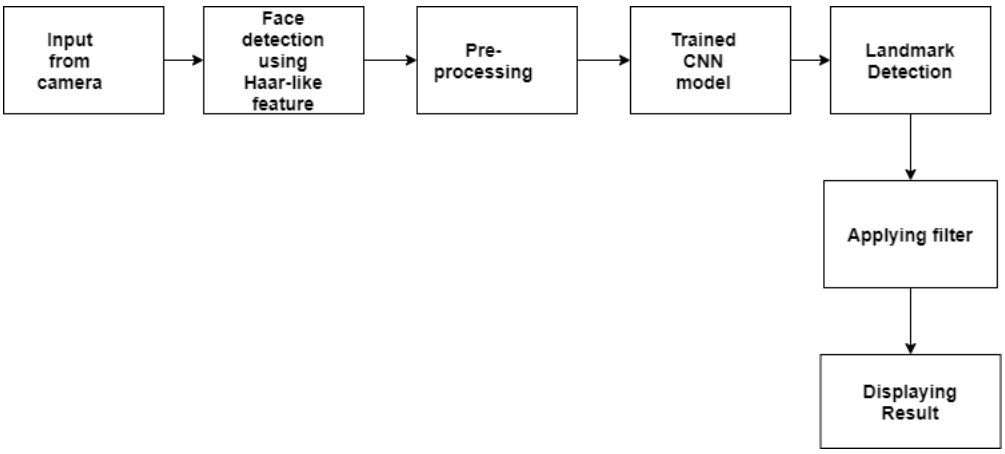


Figure 4: Block diagram of facial landmark detection using CNN

In figure 10, input scanned image from camera is taken and then passed to a face detect model which uses Haar-like features to detect faces in the given frame. The required portion of image is cropped and is normalized in the preprocessing phase which is passed to trained CNN models to detect landmarks of the face. According to the landmark, a chosen filter is applied to the face.

4.3 Pre-processing Datasets

After having image datasets, it needs to use some preprocessing to be acceptable by model. In preprocessing, both the input image and output labels should be processed.

4.3.1 Image pre-processing

Image pre-processing is the method to enhance an image or extract some useful information. For pre-processing images, a pre-processing module was used. In the preprocessing module, images were read from the directory, 3 channel images were converted into gray-scale. Then it passes to the Region of interest detector to extract the face. Crop face images were reshaped into 96 * 96 sizes and each image was normalized by dividing with 255 to convert values into float32.

4.3.2 Landmark point rescales

As only the face portion is needed to detect and face images are resized, so landmark points should also be rescale to fit in face where they need to be. It is obtained by

- By subtracting landmark point by x,y (top-most points of face detection points)
- And dividing by (weight/height of crop faces images / required dimension)

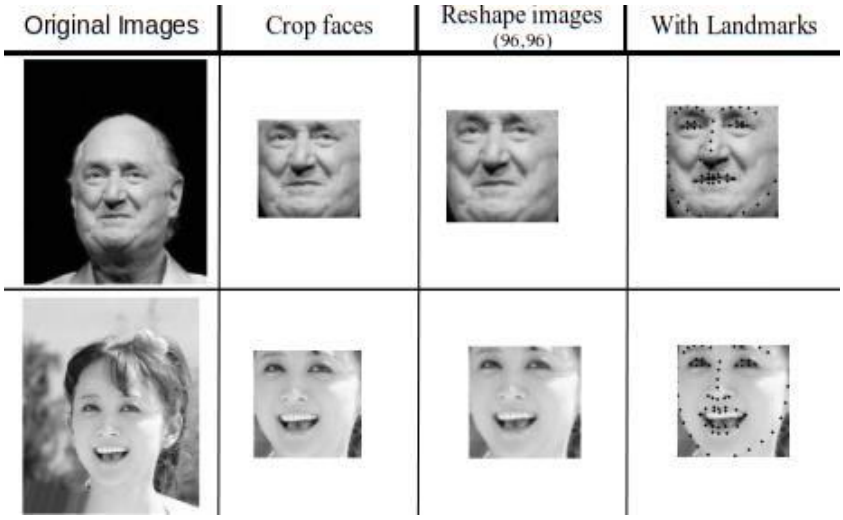


Figure 5: CelebA Dataset while preprocessing

4.4 System Description

For fully-automated facial landmark detection, the system should find Region of interest I.e., find faces of people by help of Haar-like feature algorithm and detect facial landmarks using CNN method and also large labeled data-sets to training model. So, System is divided into two parts:

4.4.1 Region of interest detector

It is the first stage of an automated land-marking system and is critical for overall performance because it extracts only those features from the data relevant for solving the defined problem such as face from the image and discards irrelevant information such as the background. For face detection, a Haar-like features method is used. All images must

first pass through a face detector to extract only the important region(s) of interest (ROI) from the data.[10]

Haar-like feature algorithm is an appearance-based approach which is used for finding the location of faces in an image by finding out which region is lighter or darker to sum up the pixel values of both regions and comparing them as the eyes region is darker than the nose region.

Haar features are used to detect the presence of feature in given image. Each feature result in a single value which is calculated by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle. Integral image concept is used to compute the rectangle features as it only needs four values at the corners of the rectangle for the calculating features. But These features are the weak classifiers. To construct a strong classifier, Adaboost is used as a linear combination of these weak classifiers. At last stage of the cascading classifier for high-performance.

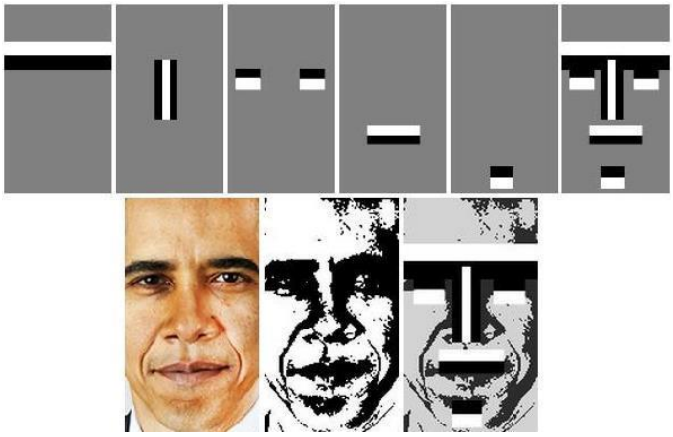


Figure 6: Haar-like feature

4.4.2 Facial landmark detector

A convolutional neural network (CNN or ConvNet) is a type of feed -forward deep learning that extracts features from the input image. It compares any image block by block and the block that it looks for in an image while detection is called as features .The convolution layer's parameters consist of a set of learnable filters.

In general the more convolutional steps we have, the more complex features (such as edges) are recognized using the proposed network. The whole process is repeated in successive layers until the system can recognize points. For example, in image classification a CNN may learn to detect edges from raw pixels in the first layer and then use the edges to detect simple shapes in the second layer. Then use these shapes to determine higher level features, such as body shapes in higher layers. [11]

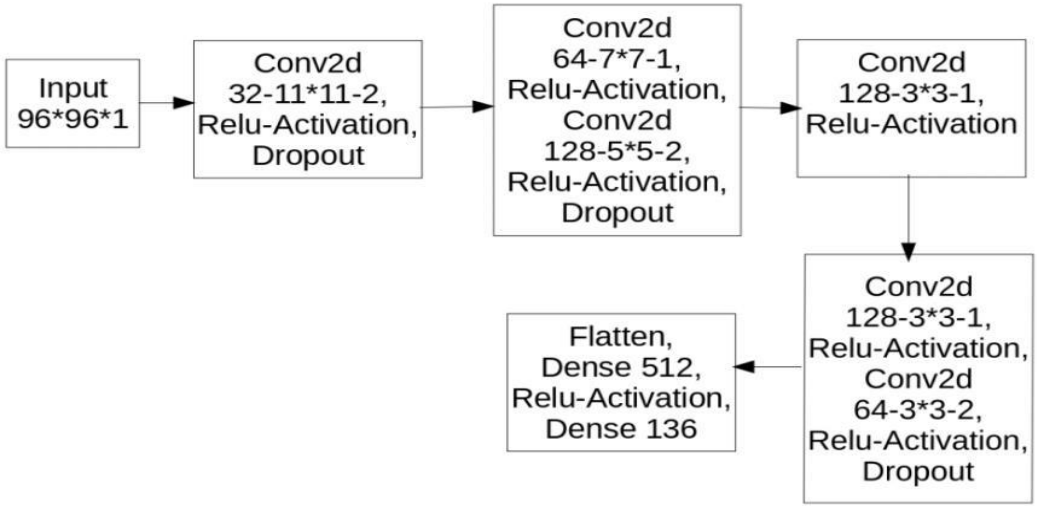


Figure 7: Architecture of proposed CNN Model

Where N-F*F-S in the figure 14 is indicated as:

N: numbers of filter

F*F: shape of filter

S: Strides

In the proposed CNN, input image is of 96 *96 *1 and was passed to convolution layers which consist of 32 numbers filters, filters shape of 11x11 and strides of 2. Then the convolutional layer was followed by ReLu Activation Layer and Dropout of 0.2. Then followed by five more convolutional layers by different numbers filters, filters shape and strides and each convolution layer was followed by ReLu Activation Layer and Dropout of 0.2. Higher level features are flattened into column-wise matrices and a fully connected 512 number of hidden neurons is used and followed by ReLu Activation Layer. At last, fully connected of 136 hidden neurons are used as outputs.

Convolutional Neural Networks that we used, have the following 3 layers: convolutional, ReLu layer, fully connected and dropout.

I) Convolutional layer

Convolution layer is the first layer to derive features from the input image. The convolutional layer conserves the relationship between pixels by learning image features using a small square of input data. It is the mathematical operation which takes two inputs such as image matrix and kernel or any filter. [12]

- The three dimensions of the image matrix is height(h)×width(w)× Color-Channel(d).
- The dimension of any filter is filter_height(fh)×filter_width(fw)×color channel(d).
- The dimension of output is height(h-fh+1)×width(w-fw+1) × 1.
- Stride is the number of pixels shifted over the input matrix.
- Sometimes the filter does not perfectly fit the input image, Pad the picture with zeros or Drop the part of the image where the filter did not fit.

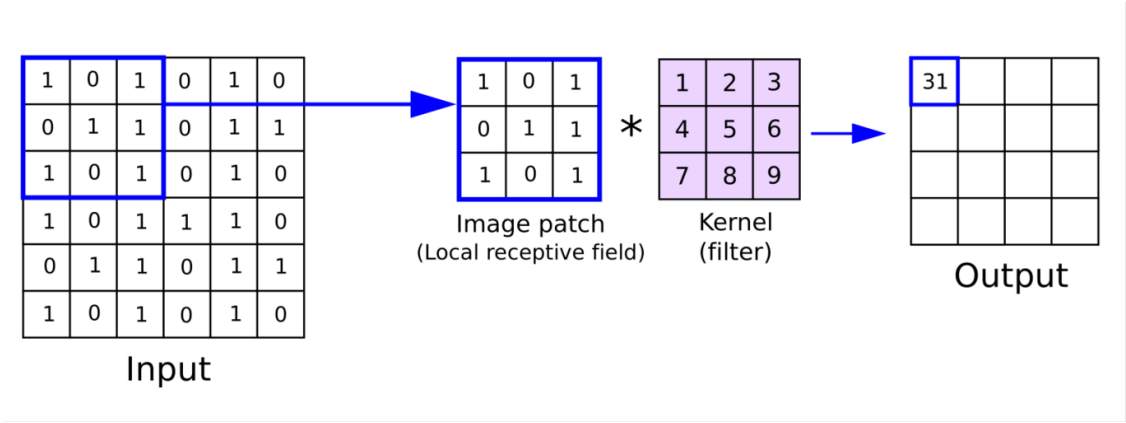


Figure 8: Convolution

II) ReLU Layer

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = \max(0, x)$. It simply removes all negative values that come from convolution layers by comparing with its function and replaces negative with zeros. ReLU's purpose is to introduce non-linearity in our ConvNet. Since the data we want our ConvNet to learn would be non-negative linear values. [13]

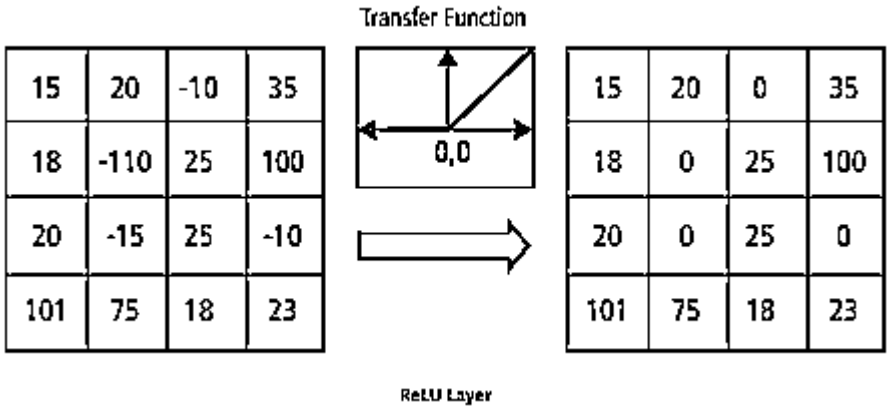


Figure 9: Rectification applied to Feature Maps

III) Fully Connected (Dense) Layer

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network where actual detection occurs by using high features to detect various landmarks based on labels.[14]

We do this by implementing the following 3 steps:

- Pick a neuron
- Pick a activation function (usually relu)
- Dot multiplies of input and neuron

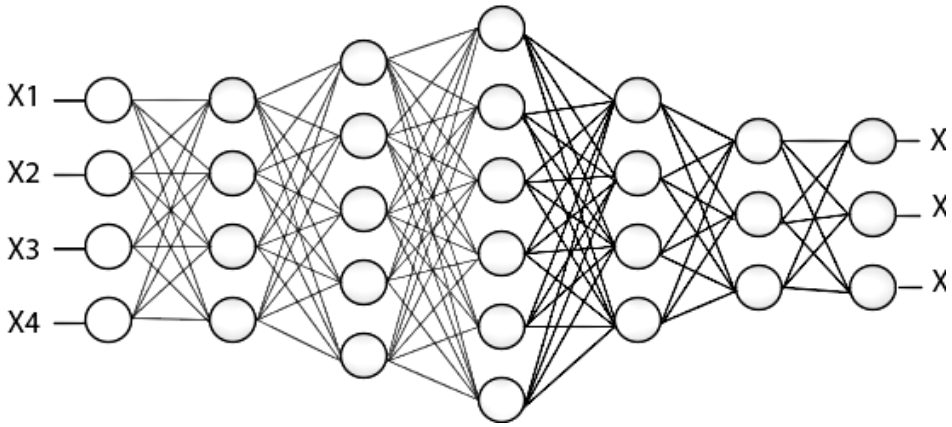


Figure 10: Fully Connected Layer

IV) Dropout

A layer that randomly sets a fraction p of the output units of the previous layer to zero. The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting .

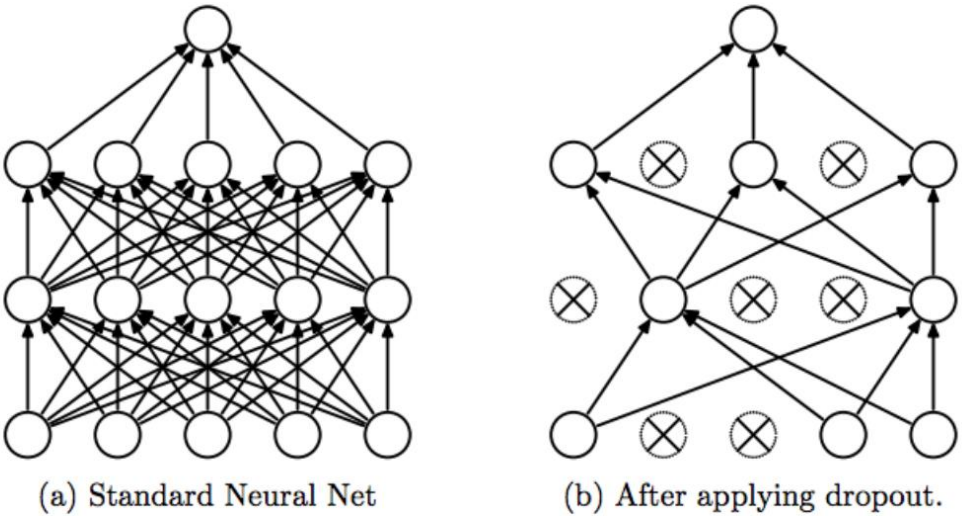


Figure 11: Dropout in action

CHAPTER 5: SYSTEM MODEL DIAGRAM

5.1 UI Diagram

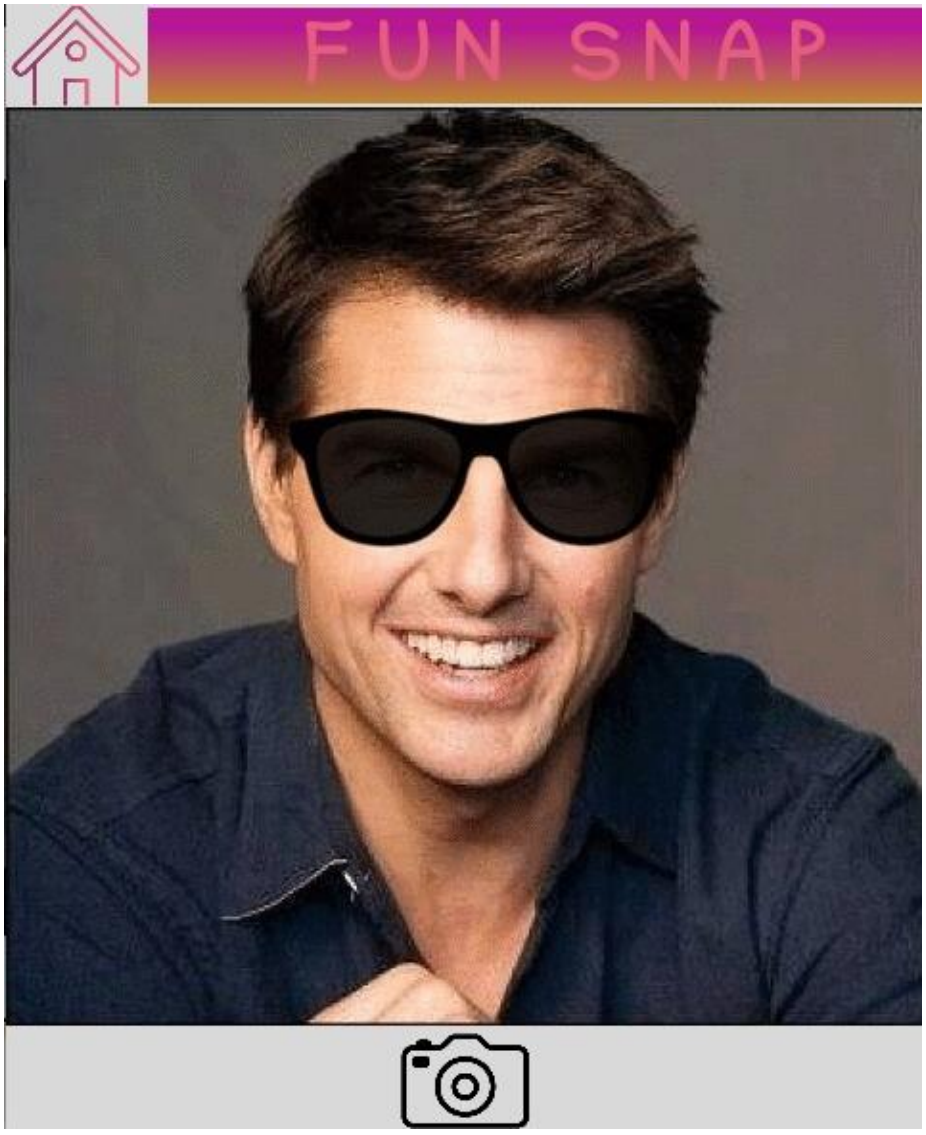


Figure 12: UI Diagram I

The figure 7 shows the GUI of facial landmark detection using CNN. It consists of a home button to return back to home. The camera button is used to open camera to detect landmark in the given frame to apply face filter.

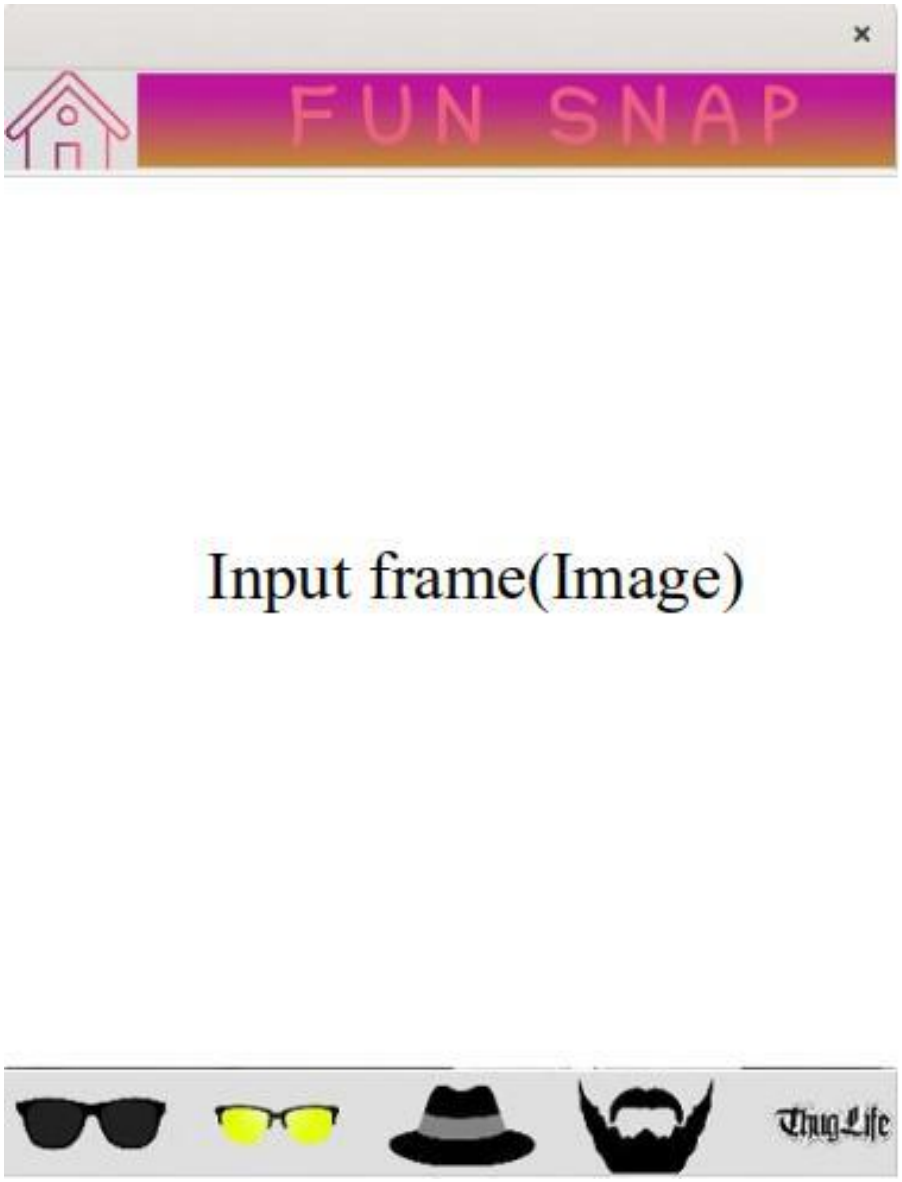


Figure 13: UI Diagram II

The figure 8 shows the GUI after the camera button has been clicked. After the frame has been uploaded to the system, it displays a detected facial landmark and the user can select a face filter which will be shown along with the user image.

5.2 Domain Model

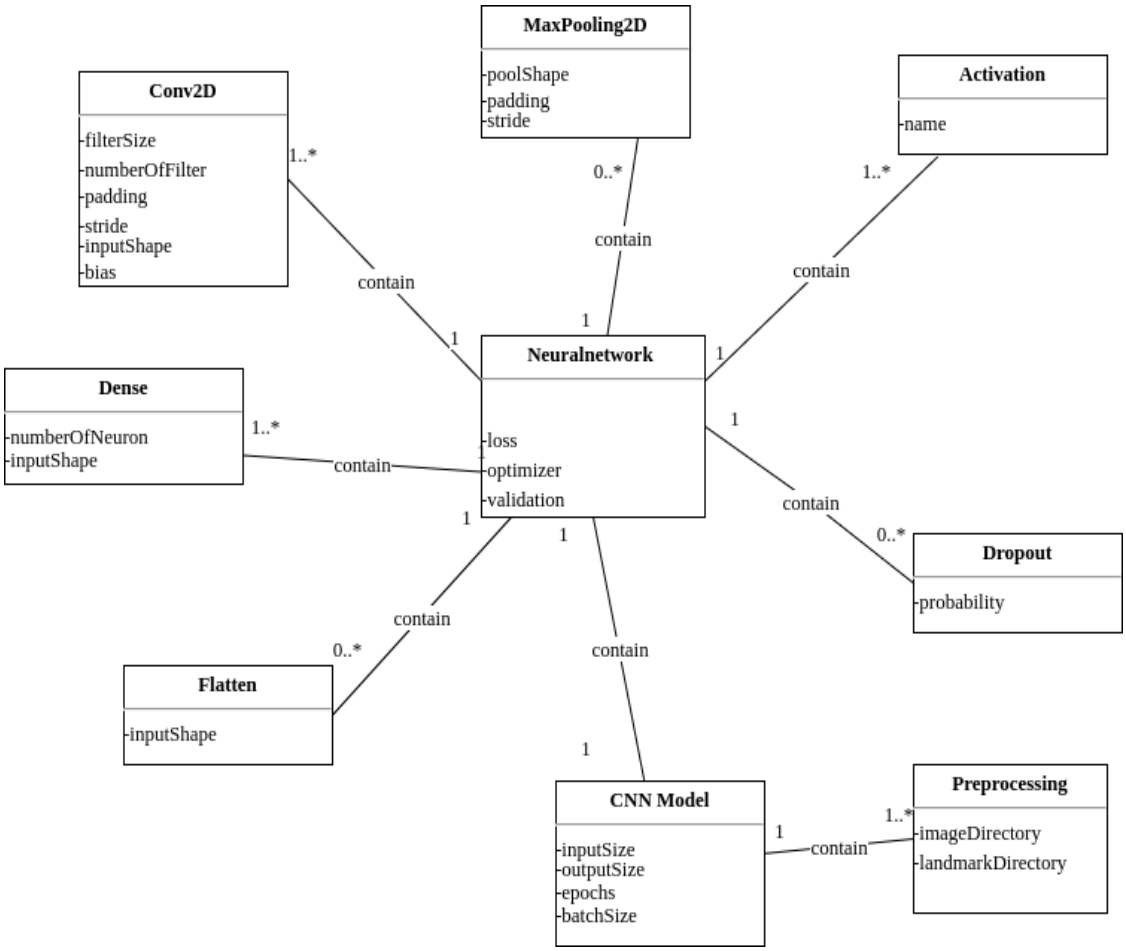


Figure 14: Domain Model of Facial Landmark Detection

The figure 5 shows the domain model of facial landmarks with CNN. The Neural network can have multiple layers of convolution, max pooling, activation layer, dense, flatten, dropout. The CNN model contains the neural network layers and it can also contain one or more than one pre-processing layer.

5.3 Class Diagram

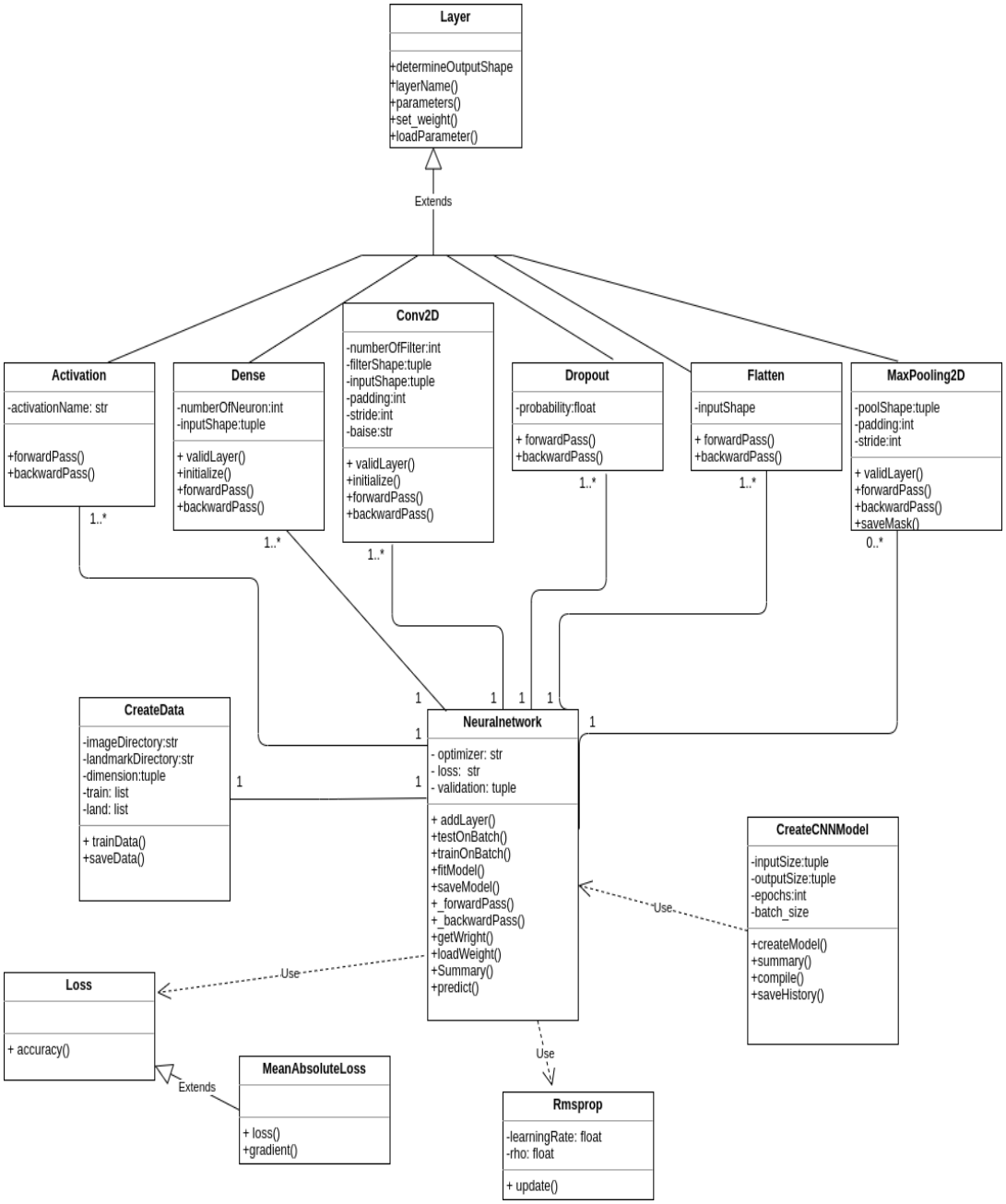


Figure 15: Class Diagram of Facial Landmark Detection

The figure 9 shows the class diagram of facial landmark detection using CNN. There are mainly 6 classes that help to build CNN model. They are conv2d, Dense, Activation, Dropout, Flatten and MaxPooling. The CreateData class is used to make train and test

data from the given data. The Neural Network class is used to build architecture of CNN model. Loss class is used to calculate the loss and it is inherited by MeanAbsoluteLoss class. Rmsprop is used to calculate the gradient to update bias and weight. CreateCNN model is used to build CNN model and to train using the train dataset.

5.4 Activity Diagram

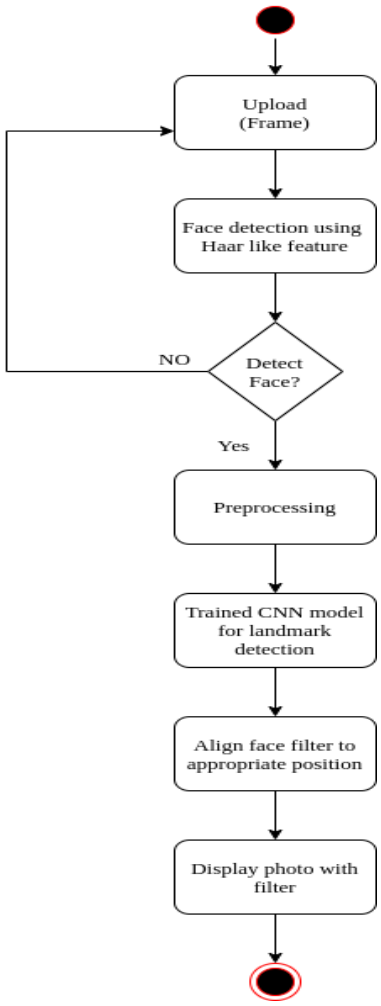


Figure 16: Activity Diagram of Facial Landmark Detection

The figure 4 shows an activity diagram of facial landmark detection using CNN model. Frames are constantly uploaded with the help of a camera. Face is detected in the frame with the help of a haar-like feature. If a face is detected in the frame it is further pre-

processed and is fed to the CNN model for outputs. The CNN model outputs 136 landmarks points and a face filter can be applied to the required position as defined in the program.

5.5 Sequence Diagram

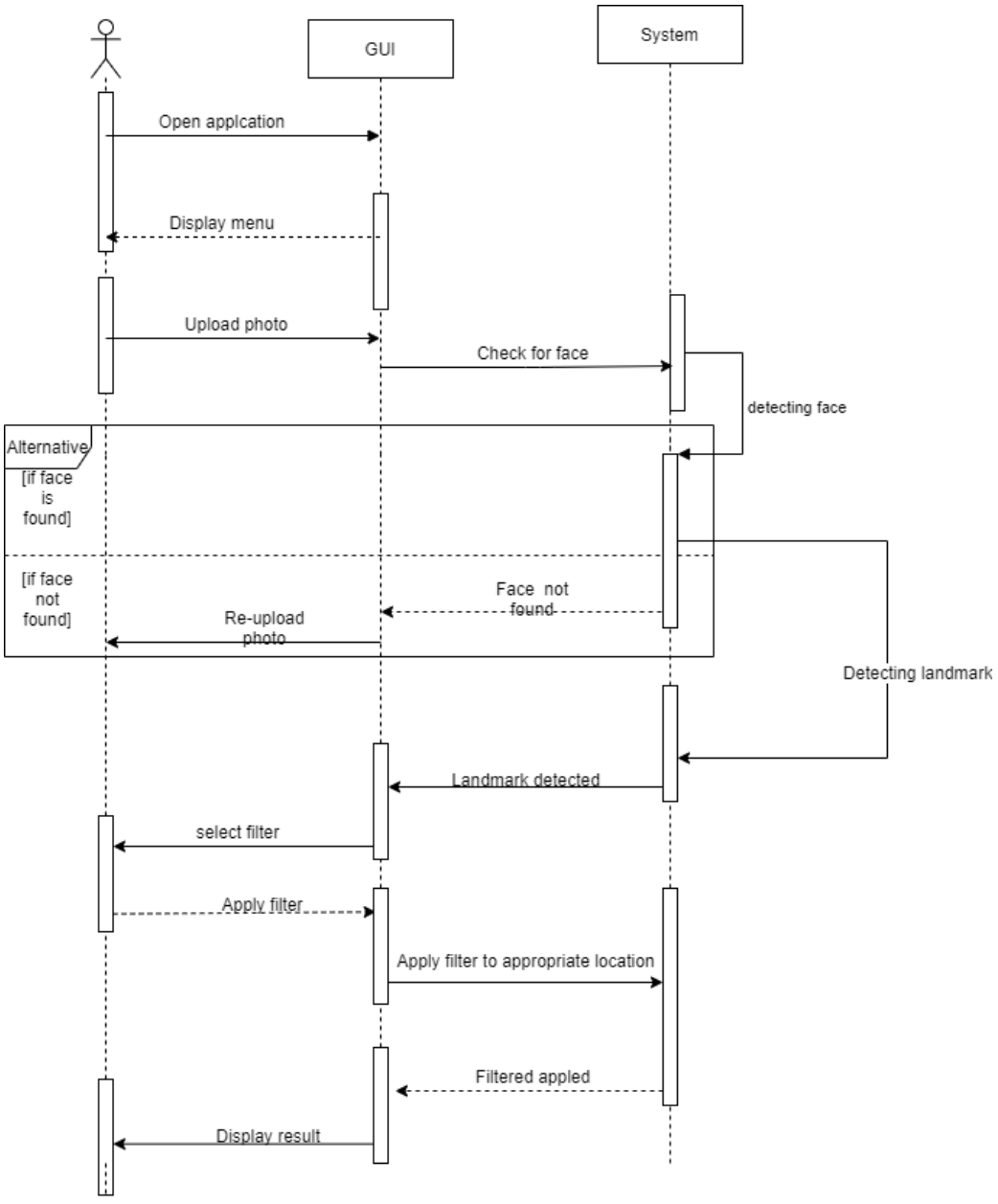


Figure 17: Sequence Diagram

The above figure shows the sequence diagram of facial landmark detection using CNN. Here users can upload frames by opening the camera in the application. The system checks for the face in the uploaded frame. If it contains the face, the system predicts 136 landmark positions. Now the user can select the face filter to be applied in the face. The GUI finally display users face with the applied filter.

CHAPTER 6: SOFTWARE DEVELOPMENT LIFECYCLE

Iterative software development model was used for developing this project. With iterative development software changes on each iteration, evolves and grows. As each iteration builds on the previous one, software design remains consistent. As requirements of this project were cleared but the CNN model needed to be built and tested for its performance as long as its accuracy was not fulfilled, an iterative development model was used.

CHAPTER 7: UNIT TESTING

For unit tests, each individual module was checked and tested for the errors. The smallest testable parts of an application, called units, were individually and independently scrutinized for proper operation. After output was verified for several inputs, we came to the conclusion the application is ok.

For unit testing, we have several inputs like Original images and their landmarks points for pre-processing modules, input shapes and other parameters for determining individual output shape, small sizes images for convolution, max-pooling and ReLu test, filters images with landmarks.

S.N	Test Cases	Test Result	Remarks
1	Convert text dataset file to dataframe(csv)	Outputs dataframe	As expected
2	Resize image and their landmark position	Resized image with their landmark position	As expected
3	Split dataset into train and test data	Train and test data	As expected
4	Shuffle dataset	Outputs shuffled data	As expected
5.	Generate batch as per given batch size of dataset	Outputs batch as given batch	As expected
6.	Check ReLU function with data	Outputs only with positive value.	As expected
7.	Check convolution with data	Outputs convoluted data.	As expected
8.	Check flatten with data	Outputs flatten data.	As expected

9	Test dropout with probability rate.	Only takes data defined by probability.	As expected
10	Test mean absolute loss with y and y_pred	Returns mean(abs(y-y_pred))	As expected

Table 2: Unit Test Case

CHAPTER 8: SYSTEM TESTING

Once all of the units in a program have been found to be working in the most efficient and error-free manner possible, larger components of the program can be evaluated by integrating all the components. The main objective of our project was to detect the landmark and after detecting landmarks to position the face filter in their right place.

S.N	Test Case	Test Result	Remarks
1	To train CNN model	Trained CNN model	As expected
2	Check whether system is detecting landmark or not	Landmarks was detected	As expected
3	Test whether system is positioning face filter in right place	Face filter was position in the right place	As expected

Table 3: System Test Case

CHAPTER 9: DISCUSSION

The project has been carried out according to the schedule successfully. Tasks like preliminary studies, detailed study, requirements analysis, feasibility study, system analysis have been completed according to the schedule. The aim is to develop a facial landmark model and to assign user selected face filters to their respective place. To achieve aim, the objective was to implement the deep neural network. CNN was used to extract landmarks from images. The implementation of CNN was done as the core task in this project. A Haar like feature was used to identify the face in the image and CNN was used to identify 136 landmark points in the face. The selection of regions of interest in input image and rescaling of landmarks was done in the preprocessing phase before it was passed through the training phase. The images were also changed into grayscale format in the preprocessing phase. 33,200 data were used for the training and 1800 were used for testing the constructed CNN model. 0.045 training loss was obtained while validation loss was about 0.05.

The obtained loss was good because of our limited resources but it could be better. The final output of this research project was good. It was able to locate the landmark but was limited to a single face. Since it runs on GUI, the performance depends upon the GPU of the machine it run on.

CHAPTER 10: CONCLUSION AND FUTURE ENHANCEMENT

10.1 Conclusion

Hence CNN was implemented in this facial landmark detection model and complete documentation, preliminary study, details study, system analysis, system design, dataset collection as well as the implementation of CNN for modeling the landmark detection was completed on time. The landmark detection CNN model was able to detect 136 landmark positions in the face. And finally, the chosen face filter can be applied to their respective places. Unlike conventional methods like SVM the developed CNN model beats them in performance.

Figure 18: training and validation loss

As shown in figure 19, we obtain training loss of 0.045 and validation loss of 0.05.

10.2 Validation

Figure 19: Validation

The above image of a person was fed to the system and the system was able to locate the landmark of the face and hence the selected filter was applied correctly.

10.3 Limitation

Although the landmark detection model was carefully planned and prepared, there are still limitations and shortcomings.

First of all, the CNN model was designed using our own code which was not optimized so that we were not able to harness the potential power of CNN. Secondly the dataset available were not prepressed to make it free from a variation in pose, lighting and background. Lastly, the model was trained on a limited dataset and limited hardware.

10.4 Future Enhancement

However, it can be further improved with addition of face recognition, facial expression analysis, 3D face modeling, age estimation, gender classification etc. It can be further

implemented in certain domains like lip reading, emotional inference, marketing, photography, etc.

REFERENCES

[1] D. Merget, M. Rock, G. Rigoll, *"Robust Facial Landmark Detection via a Fully-Convolutional Local-Global Context Network*".

[2] T. Wu, P. Turaga, R. Chellappa, "*Age estimation and face verification across aging using landmarks*".

[3] T. Devries, K. Biswaranjan, G. W. Taylor "*Multi-task learning of facial landmarks and expression*", 2014.

[4] Zhang, Shutong, and Chenyue Meng, "*Convolutional Neural Networks for Visual Recognition*".

[5] Vukadinovic, Danijela, and Maja Pantic. "*Fully Automatic Facial Feature Point Detection Using Gabor Feature Based Boosted Classifiers*".

[6] Sun, Yi, and Xiaogang Wang, "*Deep Convolutional Network Cascade for Facial Point Detection*".

[7] Martinez, Brais, and Michel F. Valstar, "*Local Evidence Aggregation for Regression Based Facial Point Detection*".

[8] Zhen-Hua Feng, Patrik Huber, Josef Kittler, William Christmas, Xiao-Jun Wu "*Random Cascaded-Regression Copse for Robust Facial Landmark Detection*".

[9] https://susanqq.github.io/UTKFace.

[10] Mita, T., Kaneko, T., & Hori, "*Joint Haar-like features for face detection*", 2005.

[11] T. Lewis, "*A Brief History of Artificial Intelligence"*, 2014, https://www.livescience.com/49007history-of artificialintelligence.html.

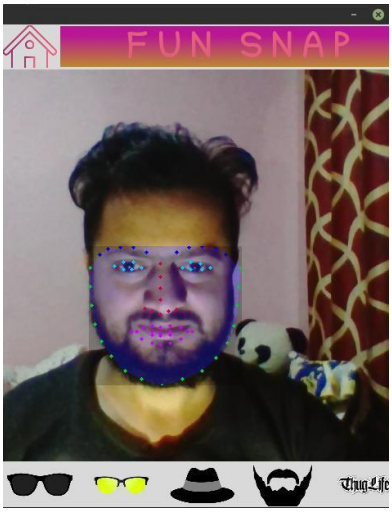
[12] https://cs.stanford.edu/people/eroberts/courses/soco/projects/neuralnetworks/History/history1

[13] P. Viola, M. J. Jones, "*Robust real-time object detection*", 2001.

[14] M. D. Zeiler and R. Fergus, "*Visualizing and understanding convolutional network*", 2013.

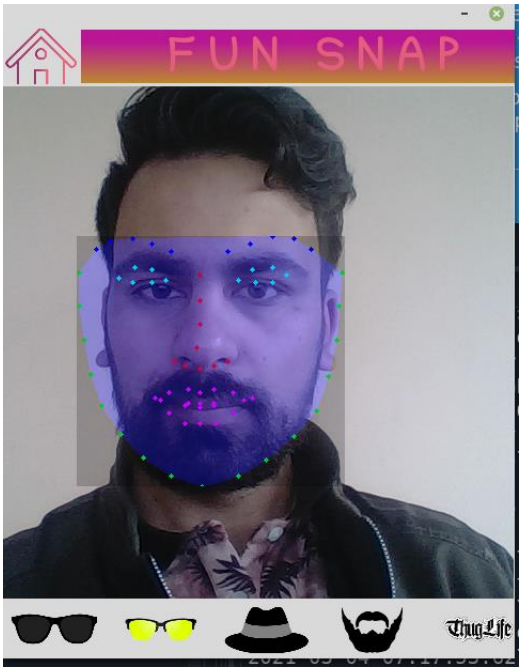
Epilogue

Output




FUN SNAP
Thug Life


FUN SNAP
Thug Life


FUN SNAP
Thug Life


FUN SNAP
Thug Life