A PRELIMINARY PROJECT REPORT ON

BE PROJECT TITLE

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Shubham Zope	Exam No: B150134351
Abhijit Watpade	Exam No: B150134350
Sanket Sonar	Exam No: B150134331
Sameer Rathod	Exam No: B150134319

DEPARTMENT OF COMPUTER ENGINEERING

K. K. Wagh Institute Of Engineering Education & Research

Hirabai Haridas Vidyanagari, Amrutdham, Panchavati, Nashik-422003
SAVITRIBAI PHULE PUNE UNIVERSITY

2020-2021

SAVITRIBAI PHULE PUNE UNIVERSITY
2020-2021

K. K. Wagh Institute Of Engineering Education & Research

CERTIFICATE

This is to certify that the Project Entitled

”SIGN TO SPEECH CONVERSION”

Submitted by

Shubham Zope	Exam No:B150134351
Abhijit Watpade	Exam No: B150134350
Sanket Sonar	Exam No: B150134331
Sameer Rathod	Exam No: B150134319

is a bonafide work carried out by Students under the supervision of Prof. N. M. Shahane and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University, for the award of Bachelor of Engineering (Computer Engineering).

Prof. N. M. Shahane
Internal Guide
Dept. of Computer Engg.

Prof. S. S. Sane
H.O.D
Dept. of Computer Engg.

Dr. K. N. Nandurkar
Principal

Place: Nashik

Date:

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary project report on ***'Sign to Speech Conversion'****.*

We would like to take this opportunity to thank our internal guide ***Prof. N. M. Shahane*** *for giving us all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful.*

We are also grateful to ***Prof. S. S. Sane****, Head of Computer Engineering Department, K. K. Wagh Institute of Engineering Education and Research for his indispensable support, suggestions.*

In the end our special thanks to ***Prof. I. Priyadarshini*** *for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for our Project.*

Shubham Zope
Abhijit Watpade
Sanket Sonar
Sameer Rathod
(B.E. Computer Engg.)

ABSTRACT

Sign language is one of the oldest and most natural form of language for communication, hence we have come up with a real time method using neural networks for finger spelling based American sign language. Automatic human gesture recognition from camera images is an interesting topic for developing vision. We propose a convolution neural network (CNN) method to recognize hand gestures of human actions from a image captured by camera. The purpose is to recognize hand gestures of human task activities from a camera image. The skin model, position of hand and orientation are applied to obtain the training and testing data for the CNN. The hand is first passed through a filter and after the filter is applied where the hand is passed through a classifier which predicts the class of the hand gestures. The hand position aims at translating and rotating the hand image to a neutral pose. Then the calibrated images are used to train the CNN.

INDEX

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

- While the world has progressed in online interaction through various tools, but the D&M (Dumb & Deaf) people still face a language barrier.So, they depend on vision based communication for interaction.
- If there is a common interface that converts the sign language to text the hand gestures can be easily understood by the other people.
- So research has been made for a vision based interface system where D&M people can enjoy the communication without really knowing each other's language.
- The aim is to develop a user friendly sign to speech conversion software where the computer understands the human sign language.
- There are different sign languages all over the world, namely American Sign Language (ASL), French Sign Language, British Sign Language (BSL), Indian Sign language, Japanese Sign Language and work has been done on other languages all around the world.

1.2 PROBLEM DEFINITION AND OBJECTIVES

To create a computer software and train a model using CNN which takes an image of hand gesture of American Sign Language and shows the output of the particular sign language in text format converts it into audio format.

1.2.0.1 **OBJECTIVES**

- create a completely functional product for the people who are not able to hear, so that ,they can get connected to world easily .
- To use and understand technologies like OpenCV, Matplotlib,Keras,Deep Learning,Python , Heroku host etc.
- To create a Web based project for detecting and understanding American sign language using Machine Learning concepts.

1.3 PROJECT SCOPE AND LIMITATION

1.4 METHODOLOGIES OF PROBLEM SOLVING

- The system is a vision based approach. All the signs are represented with bare hands and so it eliminates the problem of using any artificial devices for interaction.

- **Data Set Generation**

  For the project we tried to find already made datasets but we couldn't find dataset in the form of raw images that matched our requirements. All we could find were the datasets in the form of RGB values. Hence we decided to create our own data set. Steps we followed to create our data set are as follows.

  We have decied to use Open computer vision(OpenCV) library in order to produce our dataset.Firstly we are going to capture around 800 images of each of the symbol in ASL for training purposes and around 200 images per symbol for testing purpose. First we capture each frame shown by the webcam of our machine. In the each frame we define a region of interest (ROI) which is denoted by a blue bounded square.

  From this whole image we extract our ROI which is RGB and convert it into gray scale Image.

  Finally we apply our gaussian blur filter to our image which helps us extracting various features of our image. The image after applying gaussian blur.

  GESTURE CLASSIFICATION :-

  Algorithm Layer :-

  1. Apply gaussian blur filter and threshold to the frame taken with opencv to get the processed image after feature extraction.

  2. This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.

  3. Space between the words are considered using the blank symbol.

Activation Function :-

We have decided to use ReLU(Rectified Linear Unit) as our activation function in each layer. It is a simple function stated as **f(x) = max(0,x)** for each input pixel.Using these activation function we would be able to reduce the required computation power , Gaussian Descent losing problems, etc.

Pooling Layer :-

We apply Max pooling to the input image with a pool size of (2, 2) with relu activation function.This reduces the amount of parameters thus lessening the computation cost and reduces overfitting.

Optimizer :-

We are going to use Adam optimizer for updating the model in response to the output of the loss function. Adam combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm(ADA GRAD) and root mean square propagation(RMSProp)

Finger spelling sentence formation Implementation :-

1. Whenever the count of a letter detected exceeds a specific value and no other letter is close to it by a threshold we print the letter and add it to the current string(In our code we kept the value as 50 and difference threshold as 20).

2. Otherwise we clear the current dictionary which has the count of detections of present symbol to avoid the probability of a wrong letter getting predicted.

3. Whenever the count of a blank(plain background) detected exceeds a specific value and if the current buffer is empty no spaces are detected.

4. In other case it predicts the end of word by printing a space and the current gets appended to the sentence below.

Autocorrect Feature :-

A python library Hunspell suggest is used to suggest correct alternatives for each (incorrect) input word and we display a set of words matching the current word in which the user can select a word to append it to the current sentence.This helps in reducing mistakes committed in spellings and assists in predicting complex words.

CHAPTER 2

LITERATURE SURVEY

In the recent years there has been tremendous research done on the hand gesture recognition.
With the help of literature survey done we realized the basic steps in hand gesture recognition are :-

- Data acquisition

  The different approaches to acquire data about the hand gesture can be done in the following ways:

  - Use of sensory devices It uses electronically devices to provide exact hand configuration, and position. Different glove based approaches can be used to extract information .But it is expensive and not user friendly.
  - Vision based approach In vision based methods computer camera is the input device for observing the information of hands or fingers. The Vision Based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices. These systems tend to complement biological vision by describing 7 artificial vision systems that are implemented in software and/or hardware. The main challenge of vision-based hand detection is to cope with the large variability of human hand's appearance due to a huge number of hand movements, to different skin-colour possibilities as well as to the variations in view points, scales, and speed of the camera capturing the scene.

- Data pre processing and Feature extraction

  - In [1] the approach for hand detection combines threshold-based color detection with background subtraction.We can use Adaboost face detector to differentiate between faces and hands as both involve similar skin-color
  - We can also extract necessary image which is to be trained by applying a filter called Gaussian blur. The filter can be easily applied using open computer vision also known as OpenCV and is described in [3].

  - For extracting necessary image which is to be trained we can use instrumented gloves as mentioned in [4]. This helps reduce computation time for preprocessing and can give us more concise and accurate data compared to applying filters on data received from video extraction.
  - We tried doing the hand segmentation of an image using color segmentation techniques but as mentioned in the research paper skin color and tone is highly dependent on the lighting conditions due to which output we got for the segmentation we tried to do were no so great. Moreover we have a huge number of symbols to be trained for our project many of which look similar to each other like the gesture for symbol 'V' and digit '2', hence we decided that in order to produce better accuracies for our large number of symbols, rather than segmenting the hand out of a random background we keep background of hand a stable single color so that we don't need to segment it on the basis of skin color . This would help us to get better results.

- Gesture classification

  - In [1] Hidden Markov Models (HMM) is used for the classification of the gestures .This model deals with dynamic aspects of gestures.Gestures are extracted from a sequence of video images by tracking the skin-colour blobs corresponding to the hand into a body– face space centered on the face of the user. The goal is to recognize two classes of gestures: deictic and symbolic.The image is filtered using a fast look–up indexing table. After filtering, skin colour pixels are gathered into blobs. Blobs are statistical objects based on the location (x,y) and the colourimetry (Y,U,V) of the skin colour pixels in order to determine homogeneous areas

    [2] Naïve Bayes Classifier is used which is an effective and fast method for static hand gesture recognition. It is based on classifying the different gestures according to geometric based invariants which are obtained from image data after segmentation.Thus,unlike many other recognition

methods, this method is not dependent on skin colour. The gestures are extracted from each frame of the video,with a static background. The first step is to segment and label the objects of interest and to extract geometric invariants from them. Next step is the classification of gestures by using a K nearest neighbor algorithm aided with distance weighting algorithm (KNNDW) to provide suitable data for a locally weighted Naïve Bayes classifier.

– According to paper on "Human Hand Gesture Recognition Using a Convolution Neural Network" by Hsien-I Lin , Ming-Hsiang Hsu, and Wei-Kai Chen graduates of Institute of Automation Technology National Taipei University of Technology Taipei, Taiwan, they construct a skin model to extract the hand out of an image and then 9 apply binary threshold to the whole image. After obtaining the threshold image they calibrate it about the principal axis in order to center the image about it. They input this image to a convolutional neural network model in order to train and predict the outputs. They have trained their model over 7 hand gestures and using their model they produce an accuracy of around 95% for those 7 gestures.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 ASSUMPTION AND DEPENDENCIES

3.2 FUNCTIONAL REQUIREMENT

3.2.1 System Input

System will take input as image.Image will consist the sign.

3.2.2 System Output

System will give appropriate word corresponding to the signs

3.3 NON FUNCTIONAL REQUIREMENTS

3.3.1 Performance Requirements

The System should be able to capture the hand signs in considerable time. It should give the right output every-time.

3.3.2 Safety Requirements

The System must be reliable. It should take care of the maximum capacity of handling the hand signs. The system should not crash in any circumstances and show uniform behavior in all cases.

3.3.3 Software Quality Attributes

The system should have QA attributes like adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability.

3.4 SYSTEM REQUIREMENTS

3.4.1 Database Requirements

Files system

3.4.2 Software Requirements (Platform Choice)

Google Colab

3.4.3 Hardware Requirements

External/Internal Camera

3.5 ANALYSIS MODELS : INCREMENTAL MODEL

SDLC Model to be applied is Incremental model. The incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements.

CHAPTER 4

SYSTEM DESIGN

4.1 DATA FLOW DIAGRAM

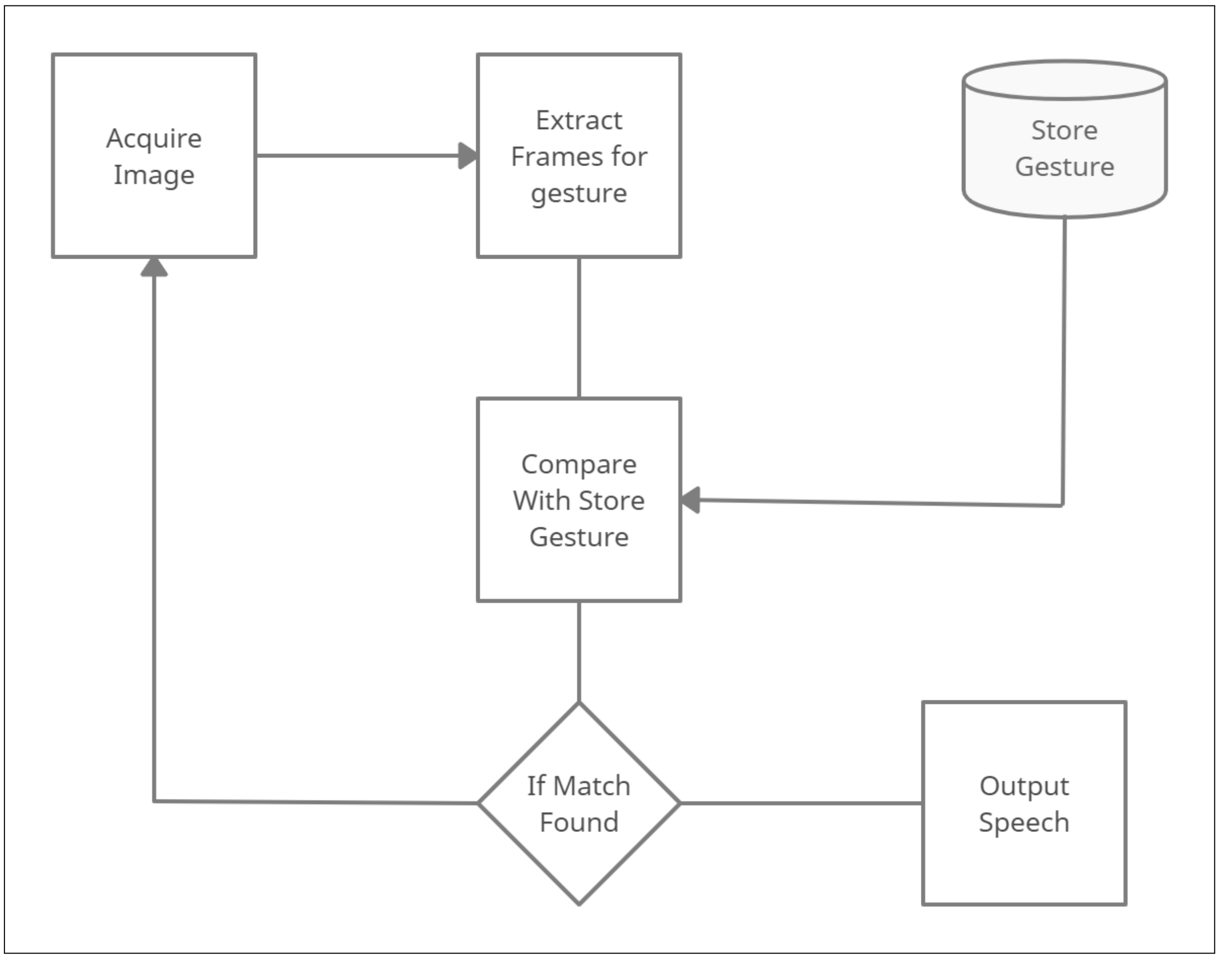

Figure 4.1: Data Flow

4.2 UML DIAGRAM

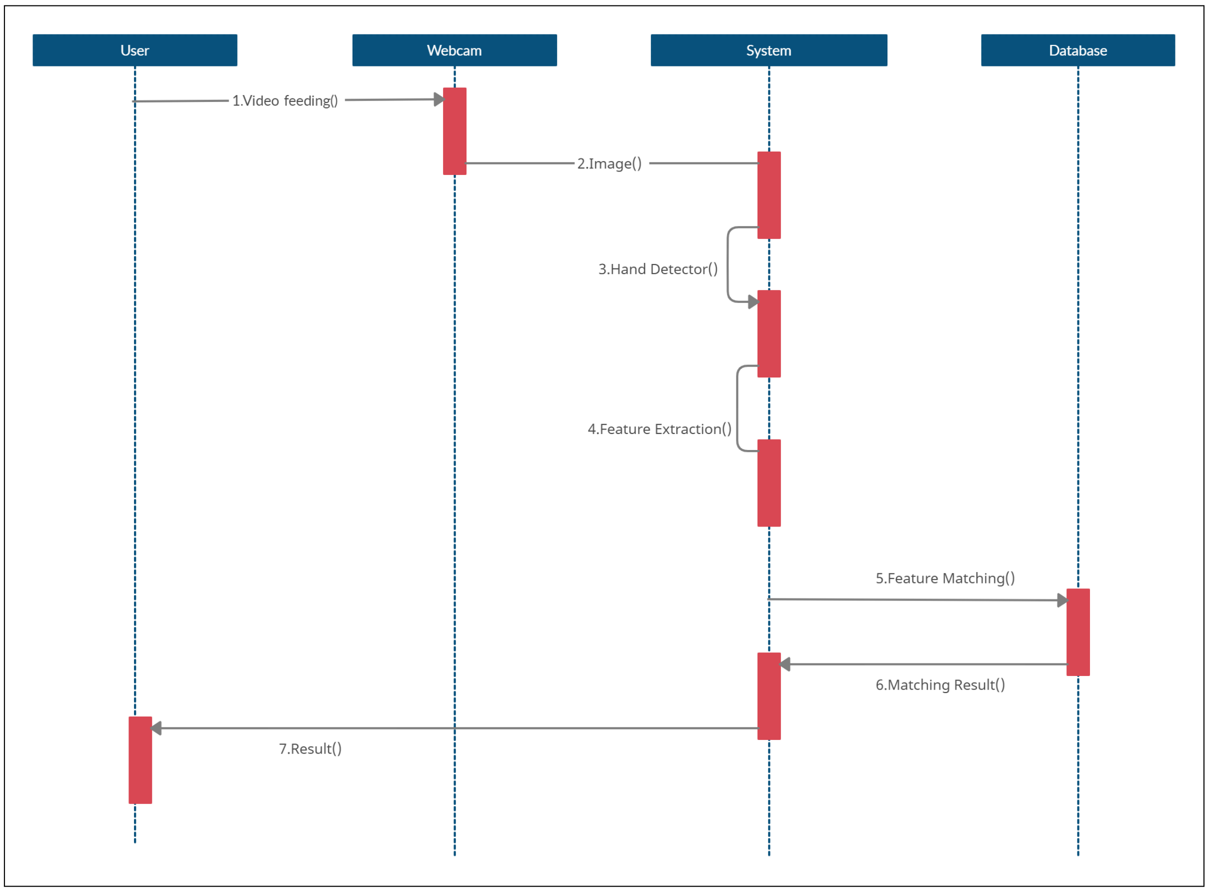

Figure 4.2: Sequence Diagram

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATE

5.1.1 Reconciled Estimate

5.1.1.1 Cost Estimates

With an aim to develop the project using open source tools, no major cost consuming factors exist.

5.1.1.2 Time Estimates

The estimated time of our project is 8 months.

5.1.2 Project Resources

- People: Internal Guide, External Guide, Development Team
- Hardware: CPU, GPU, External/Internal Camera
- Software: Windows OS, Python Libraries

5.2 RISK MANAGEMENT

5.2.1 Risk Identification

Risk Identification forms an important part in Software Development Life Cycle. This gives a view of all the possible risks to the system and whether they can be controlled or not. We may define risks by presenting a questionnaire and answering all the questions relating to it or we also have a table giving risk details and probability of occurrence.

1. Are end-users enthusiastically committed to the project and the system/product to be built? Yes, they will be enthusiastic about using this project since it is very helpful as it helps them to form communication with others.
2. Are requirements fully understood by the software engineering team and its customers? Yes

3. Have customers been involved fully in the definition of requirements? Yes, we are building this project keeping the end user in mind.
4. Do end users have realistic expectations? Yes
5. Does the software engineering team have the right mix of skills? Yes
6. Are the project requirements stable? Yes
7. Is the number of people working on the project adequate to do the job? Yes

5.2.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Underestimation of time required to train the model	Low	High	Medium	High
2	No datasets available	High	Medium	High	High

Table 5.1: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 5.2: Risk Probability definitions

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 5.3: Risk Impact definitions

Risk ID	1
Risk Description	Underestimation of time required to train the model
Category	Technology
Source	Identified during early development
Probability	Low
Impact	High
Response	Accept
Strategy	Use of Google Colab Platform for training
Risk Status	Identified

Risk ID	2
Risk Description	No datasets available
Category	Technology
Source	Identified during early development
Probability	High
Impact	High
Response	Mitigate
Strategy	Create your own dataset.
Risk Status	Identified

5.3 PROJECT SCHEDULE

5.3.1 Project Task Set

Major Tasks in the Project Stages are:

- Task 1 (T1): Creating the dataset.
- Task 2 (T2): Learning and understanding basics of machine learning.
- Task 3 (T3): Cleaning and pre-processing of the dataset.
- Task 4 (T4): Selection of highest accuracy algorithm for training the model.
- Task 5 (T5): Creating a client terminal for using the trained model and make various predictions.

5.3.2 Task Network

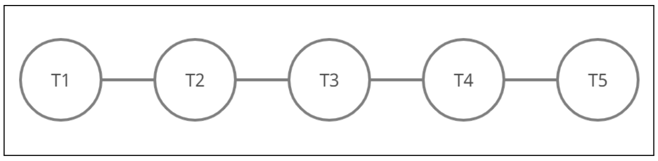

Figure 5.1: Task Network

5.4 TEAM ORGANIZATION

5.4.1 Team Structure

Team consists of 4 members and 1 internal guide

5.4.2 Management Reporting and Communication

There are reviews which take place within the semester . Changes and modifications are suggested in these reviews and acted upon later. Communication takes place through:

- Email
- Whatsapp
- Google Meet

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

The system is a vision based approach. All the signs are represented with bare hands and so it eliminates the problem of using any artificial devices for interaction.

1. **Data-set Generation:-**

   For the project we tried to find already made datasets but we couldn't find dataset in the form of raw images that matched our requirements. All we could find were the datasets in the form of RGB values. Hence we decided to create our own data set. Steps we followed to create our data set are as follows.

   We used Open computer vision(OpenCV) library in order to produce our dataset.Firstly we captured around 700 images of each of the symbol in ASL for training purposes and around 300 images per symbol for testing purpose.

   First we capture each frame shown by the webcam of our machine. In the each frame we define a region of interest (ROI) which is denoted by a blue bounded square as shown in the image below.

   From this whole image we extract our ROI which is RGB and convert it into gray scale Image as shown below.

   Finally we apply our gaussian blur filter to our image which helps us extracting various features of our image.

2. **Gesture Classification** ;-

   - **Layer 1 ;- CNN Model**

     (a) **1st Convolution Layer :-** The input picture has resolution of 128x128 pixels.It is first processed in the first convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 126X126 pixel image, one for each Filter-weights.

     (b) **1st Pooling Layer :-** The pictures are down sampled using max pooling of 2x2 i.e we keep the highest value in the 2x2 square of array. Therefore, our picture is down sampled to 63x63 pixels.

(c) **2nd Convolution Layer :-** Now, these 63 x 63 from the output of the first pooling layer is served as an input to the second convolutional layer.It is processed in the second convolutional layer using 32 filter weights (3x3 pixels each).This will result in a 60 x 60 pixel image.

(d) **2nd Pooling Layer :-** The resulting images are downsampled again using max pool of 2x2 and is reduced to 30 x 30 resolution of images.

(e) **1st Densely Connected Layer :-** Now these images are used as an input to a fully connected layer with 128 neurons and the output from the second convolutional layer is reshaped to an array of 30x30x32 = 28800 values. The input to this layer is an array of 28800 values. The output of these layer is fed to the 2nd Densely Connected Layer.We are using a dropout layer of value 0.5 to avoid overfitting.

(f) **2nd Densely Connected Layer :-** Now the output from the 1st Densely Connected Layer are used as an input to a fully connected layer with 96 neurons.

(g) **Final layer :-** The output of the 2nd Densely Connected Layer serves as an input for the final layer which will have the number of neurons as the number of classes we are classifying (alphabets + blank symbol).

3. **Testing and Training :-** We convert our input images(RGB) into grayscale and apply gaussian blur to remove unnecessary noise.We apply adaptive threshold to extract our hand from the background and resize our images to 128 x 128.

   We feed the input images after preprocessing to our model for training and testing after applying all the operations mentioned above.

   The prediction layer estimates how likely the image will fall under one of the classes. So the output is normalized between 0 and 1 and such that the sum of each values in each class sums to 1. We have achieved this using softmax function.

At first the output of the prediction layer will be somewhat far from the actual value. To make it better we have trained the networks using labeled data. The cross-entropy is a performance measurement used in the classification. It is a continuous function which is positive at values which is not same as labeled value and is zero exactly when it is equal to the labeled value. Therefore we optimized the cross-entropy by minimizing it as close to zero. To do this in our network layer we adjust the weights of our neural networks. TensorFlow has an inbuilt function to calculate the cross entropy.

As we have found out the cross entropy function, we have optimized it using Gradient Descent in fact with the best gradient descent optimizer is called Adam Optimizer.

4. **Model Deployment :-** Going to deploy our model using Tkinter.

6.2 TOOLS AND TECHNOLOGY USED

- Python for implementation.
- Google colab for model creation.
- Adam optimizer for fast Gradient Decent calculations and adjusting the learning rate effectively.
- ImageDataGenerator for creating the wide verity of data from existing data with data augmentation.
- Open-CV for sign language image collection
- Keras for Deep Learning
- TKinter for frontend

6.3 ALGORITHM DETAILS

1. **Convolutional neural networks :-** Unlike regular Neural Networks, in the layers of CNN, the neurons are arranged in 3 dimensions: width, height, depth. The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner. Moreover, the final output layer would have dimensions (number of classes), because by the end of the CNN architecture we will reduce the full image into a single vector of class scores.

   (a) **Convolution Layer :-** convolution layer we take a small window size [typically of length 5*5] that extends to the depth of the input matrix. The layer consist of learnable filters of window size. During every iteration we slid the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position. As we continue this process well create a 2-Dimensional activation matrix that gives the response of that matrix at every spatial position. That is, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color

   (b) **Max Pooling Layer :-** max pooling we take a window size [for example window of size 2*2], and only take the maximum of 4 values. Well lid this window and continue this process, so well finally get a activation matrix half of its original Size.

   (c) **Final Output Layer :-** In convolution layer neurons are connected only to a local region, while in a fully connected region, well connect the all the inputs to neurons.

   (d) **Final Output Layer :-** After getting values from fully connected layer, well connect them to final layer of neurons[having count equal to total number of classes], that will predict the probability of each image to be in different classes.

2. **Optimizer :-** We have used Adam optimizer for updating the model in response to the output of the loss function. Adam combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm(ADA GRAD) and root mean square propagation(RMSProp)

CHAPTER 7

SOFTWARE TESTING

7.1 TYPES OF TESTING

7.2 TEST CASES AND TEST RESULTS

CHAPTER 8

RESULTS

CHAPTER 9

CONCLUSIONS

9.1 CONCLUSION

In this report, a functional real time vision based American sign language recognition for deaf and dumb people have been developed for ASL alphabets.We can achieve final accuracy of 98.74 percent on our data set. We would be able to improve our prediction after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other. This way we are able to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate.

9.2 FUTURE WORK

- We can make the application completely hands free using the Open-CV so that the user would not require to use the mouse and keyboard again and again.
- We can provide support of multiple language using the techniques of natural language processing.
- If in case application could not be able to perform in certain conditions than we can have a wide community of people as volunteer the conversion of sign language live.
- We can provide a support for the words that can be defined with the help of stream of signs rather than a single sign at a time.
- As our application uses the technique of finger spelling for sign language we can upgrade our application so that it can support other form sign languages as well.

9.3 APPLICATIONS

- It can provide an easy conversation between dumb or deaf people and the other people.
- It can be used to teach new learner of the sign languages easily and effectively, in more faster way.
- It can greatly help in the effective education of dumb or deaf people so that they can learn and connect with the outer world easily.
- As the application is more portable ,it can be used anywhere and at anytime in the world with out requirement of the internet being offline.

ANNEXURE A

REFERENCES

Thomas Noltey, Hans Hanssony, Lucia Lo Belloz,”Communication Buses for Automotive Applications” In Proceedings of the 3rd Information Survivability Workshop (ISW-2007), Boston, Massachusetts, USA, October 2007. IEEE Computer Society.