Mini Project Report on

Sign Language Recognition using CNN

by

Pranit Jadhav (19CE5502)

Kunal Kamble (19CE5503)

Atharva Shanware (18CE7013)

Under the guidance of

Mrs Pallavi Chitte

Department of Computer Engineering

Ramrao Adik Institute of Technology

Dr. D. Y. Patil Vidyanagar, Nerul, Navi Mumbai

University of Mumbai

May 2021

Ramrao Adik Institute of Technology

Dr. D. Y. Patil Vidyanagar, Nerul, Navi Mumbai

CERTIFICATE

This is to certify that Mini Project report entitled

Sign Language Recognition using CNN

by

Pranit Jadhav (19CE5502)

Kunal Kamble (19CE5503)

Atharva Shanware (18CE7013)

is successfully completed for Third Year Computer Engineering as prescribed by University of Mumbai.

Supervisor	Project Coordinator
(Mrs Pallavi Chitte)	(Name of Project Coordinator)
Head of Department	Principal
(Dr. Leena Ragha)	(Dr. Mukesh D. Patil)

Mini Project Report Approval

This is to certify that the Mini Project entitled "**Sign Language Recognition using CNN** " is a bonafide work done by **Pranit Jadhav , Kunal Kamble and Atharva Shanware** under the supervision of **Mrs. Pallavi Chitte**. This Mini Project has been approved for Third Year Computer Engineering.

Internal Examiner :

1. .....

2. .....

External Examiners :

1. .....

2. .....

Date : . . . /. . . /. . . . . .

Place : . . . . . .

DECLARATION

I declare that this written submission represents my ideas and does not involve plagiarism. I have adequately cited and referenced the original sources wherever others' ideas or words have been included. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will cause disciplinary action against me by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: _____

Pranit Jadhav (19CE5502)

Kunal Kamble (19CE5503)

Atharva Shanware (18CE7013)

Abstract

In a recent survey by World federation for Deaf (WFD) , it was analysed that over the total population in the world 5% are having hearing and speaking disability in which approximately adults and children are 328 million and 32 million respectively. These people communicate with the hand gestures which are not known to the common people so here develops a barrier between communication with D&M people.To solve this problem Sign Language Recognition was developed so that computer is able to perform conversion of hand gestures into text by ASL fingerspelling .These Softwares use various Machine Learning Models to perform conversions. The Software requires two types of users; user 1 is a D & M person and user 2 can be a common person or D&M person and user 1 performs hand gestures in a specific window of the webcam and user 2 gets the output in form of character,word or sentence. In case the user 2 understands other languages like chinese, japanese or else, in that case software has a translator that converts text into various languages with spoken words. The purpose of this software is to make dynamic communication with D&M people.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

There are plenty of people in the world with speech and hearing defects. To overcome this limitation they communicate through the medium of hand gestures which is also known as sign language. Yet there is a huge communication gap as the sign language is still unknown to a large amount of the population and each region has their own sign language based on their native language. This problem of communication gap can be easily solved through computer vision and machine learning. This Document represents Sign Language Recognition Software by using Convolution Neural Network that converts the ASL fingerspelling gestures into text which can be translated into different languages.

Software consists of three parts :

- Obtaining video of the user signing (input).
- Converting each frame in the input of the webcam to a character.
- Conversion of hand gestures into words from classification scores (output).

1.2 Objectives

The Sign Language Recognition Software is developed to solve the problem of communication between the people that are unable to hear and speak.The problem occurs because these people communicate with help of hand gestures that are not known to the normal people.So with the help of Sign Language Recognition Software, hand gestures performed by the D&M people are converted into text and also spoken words. The Software also provides a feature to convert the text into various other languages like chinese,japanese and many more. In order to address this problem and to perform dynamic communication, we present a sign language recognition system that uses Convolutional Neural Networks (CNN) to translate a video of a user’s ASL signs fingerspelling into text.

1.3 Motivation

The number of deaf-mutes in the country are roughly calculated between 1.8 million and 7 million. (The wide range in population estimates exists because the Indian census doesn't track the number of deaf people — instead, it documents an aggregate number of people with disabilities).Deaf and Mutes (D&M) people, inability to hear and speak.They have a totally different culture compared to other people . They are more smart and creative. Their culture mainly focuses on social beliefs like art, history, dance and many more. They communicate through sign language (hand gestures). So communication has become a difficult task. To solve the communication barrier we present this software.Sign language substantially facilitates communication to people with such disabilities.

1.4 Organization of report

Chapter 1 : The introduction and the need of the project is discussed.

Chapter 2 : The survey of the existing system, and the research work carried out by various people.

Chapter 3 : The proposed model of the project, its design and implementation.

Chapter 4 : The results section which contains screenshots and explains the functioning of the working computer model.

Chapter 5 : conclusion of the project and also, some scope of future works in the same space.

Chapter 2

Literature Survey

2.1 Existing Systems

Sign Language Recognition is a very influential topic and in recent years it has been gaining height of increasing interest.

- Tremendous type of research has been performed with various contributions.
- Over the years, different researchers tried to solve this problem including Bayesian networks, linear classifiers and neural networks
- Singha and Das obtained accuracy of 96% on 10 classes for images of gestures of one hand using Karhunen-Loeve Transforms.
- Special webcam Microsoft Kinect developed by Microsoft has been used in such projects due to its skin masking properties which has a greater accuracy.

Sr no.	Researchers	Research Paper	Methodology
1.	Archana S Ghotkar, Rucha Khatal, Sanjana Khupase, Surbhi Asati and MIthila Hadop	Hand Gesture Recognition for Indian Sign Language	Camshift and HSVmodel using Genetic algorithm
2.	P Subha Rajan and Dr G Balakrishnan	Sign Language Recognition for deaf and dumb people	7 bit orientation and generation process through RIGHT and LEFT scan
3.	T. Shanableh	Arabic sign language gestures	K-NN and polynomial networks

Table 2.1.1: Survey of existing systems

2.2 Limitations of Existing System

- Most of the existing systems require a High-Tech glove with motion sensors to capture the gesture in 3 dimensions or even a Microsoft Kinect, both of which are pretty expensive.
- These also impose scalability issues due to the equipment dependency.
- Benefits of portability have to be compromised, with the equipment having to be carried everywhere where the system is to be used.
- Software is costly and difficult to use commercially.

Chapter 3

Proposed System

3.1 Problem Statement

The goal of this problem statement is to develop a system that captures the image through the live webcam only if a particular gesture is present and give the text or letter associated with the gesture. The system shall accept input of a static gesture through the webcam, preprocess the image, feed it to the CNN model and display the text as an output. The system only gives output for the ASL fingerspelling. Our problem consists of three tasks to be done in real time:

1. Obtaining video of the user signing (input).
2. Classifying each frame in the video to a letter.
3. Reconstructing and displaying the most likely word from classification scores (output).

From a computer vision perspective, this problem represents a significant challenge due to a number of considerations, including:

- Environmental concerns (e.g. lighting sensitivity, background, and camera position)
- Occlusion (e.g. some or all fingers, or an entire hand can be out of the field of view)
- Sign boundary detection (when a sign ends and the next begins)
- Coarticulation (when a sign is affected by the preceding or succeeding sign)

The system shall recognize the specified letter through the webcam. It shall also form words and sentences using custom gestures. Then after confirming the sentence it will convert the entire sentence into speech, and also translate the sentence in the specified language and that into speech too. The aim of this problem statement is to close the communication gap faced by the people with hearing and speech disability. The proposed system is limited to static gestures which can only recognize ASL fingerspelling but can be later updated for ISL, JSL and also dynamic hand gesture and two hand gestures.

3.2 Proposed Methodology/Techniques

This is a vision based system, it uses only a webcam for its functioning. This eliminates the use of equipment such as flex sensors, kinect, etc for interaction. So all the user has to do is place the gestures in front of the camera.

Data Preprocessing:

Before feeding the data to the CNN model it is necessary to preprocess the data. This enables the CNN model to recognize the labels more accurately. There is also the need to eliminate the background, as it can cause hindrance to the recognition system. So the data preprocessing takes place as follows:

- Set borders: The entire image is not considered for recognition; only a part where the hand is supposed to be placed is cropped.
- Skin Masking: Skin masking is applied so that only the hand is visible and all the other background noises are eliminated. The cropped image is converted to HSV and the histogram of that image is calculated. The histogram is used to find the features of the image using an opencv function calcBlackProject(). This function is used to find the features of the image, in this case it is used to find the flesh color areas in the image.

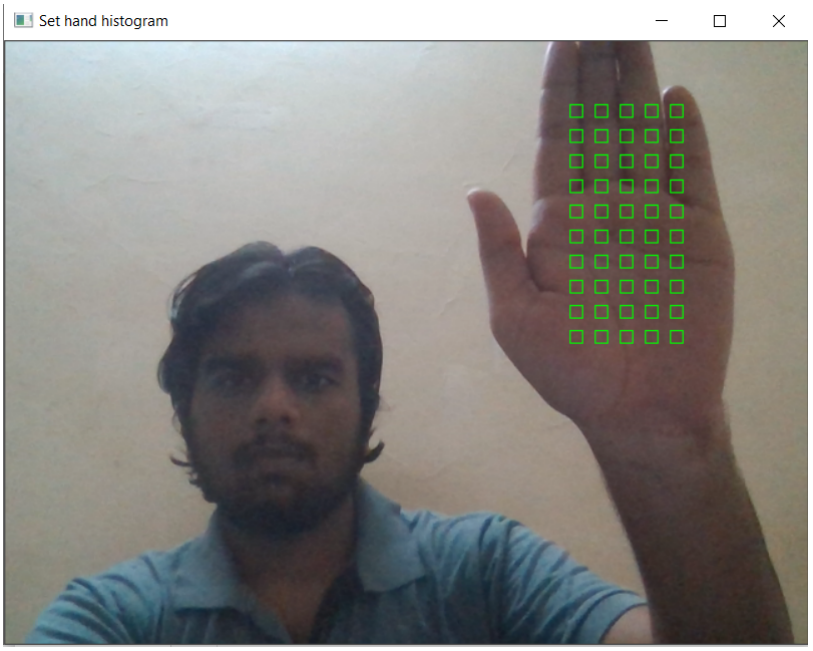

Fig 3.2.1: Input image

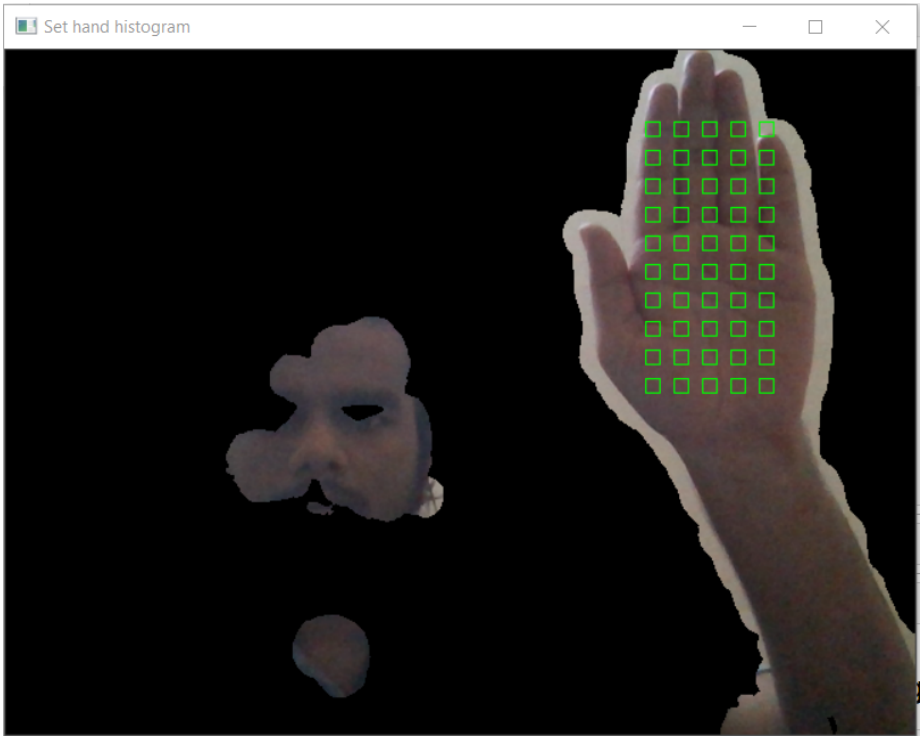

Fig 3.2.2 Skin masking

- Removing noise and Smoothening:
    - Adaptive Thresholding: In simple thresholding, the threshold value is global, i.e., it is the same for all the pixels in the image. Adaptive thresholding is the method where the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions. In OpenCV, you can perform Adaptive threshold operation on an image using the method adaptiveThreshold().
    - Gaussian Blur: Gaussian blur is applied to the image which helps in extracting various features of image from ROI. ADAPTIVE_THRESH_GAUSSIAN_C: The threshold value is a gaussian-weighted sum of the neighbourhood values minus the constant C.

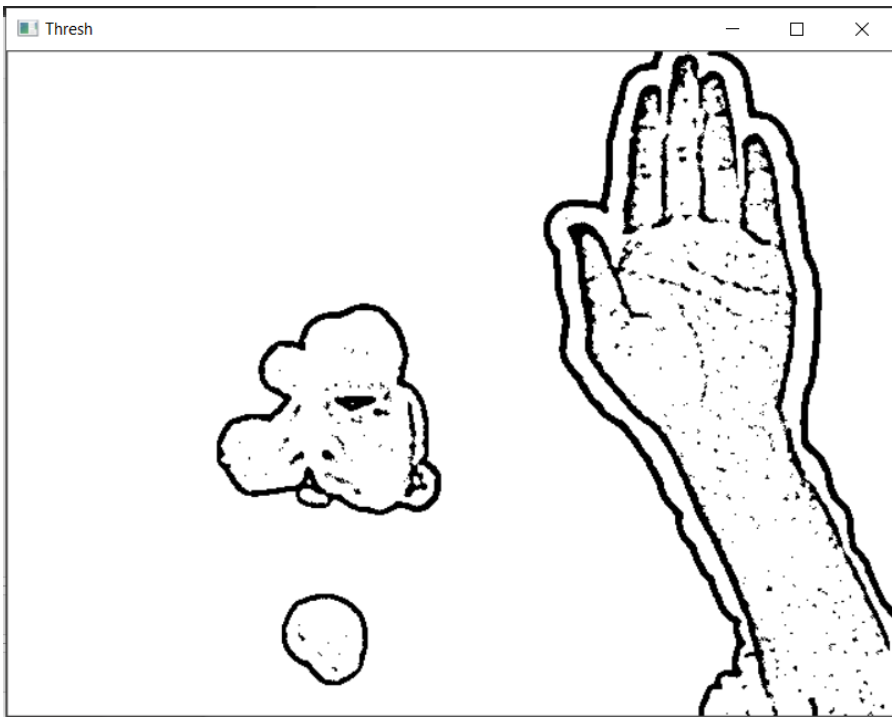

Fig 3.2.3: Gaussian Blur

- Convolutional Neural Network Model:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The CNN model is fed with the preprocessed dataset of ASL fingerspelling. The architecture used for the CNN model consists of 32, 64, 64, 128 bit architecture, the filter used in the convolution layer is of size 5x5 in the first layer and 3x3 in the rest. Max pooling is also applied with the filter size of 3x3 in the first layer and 2x2 in second.

- CNN Model:

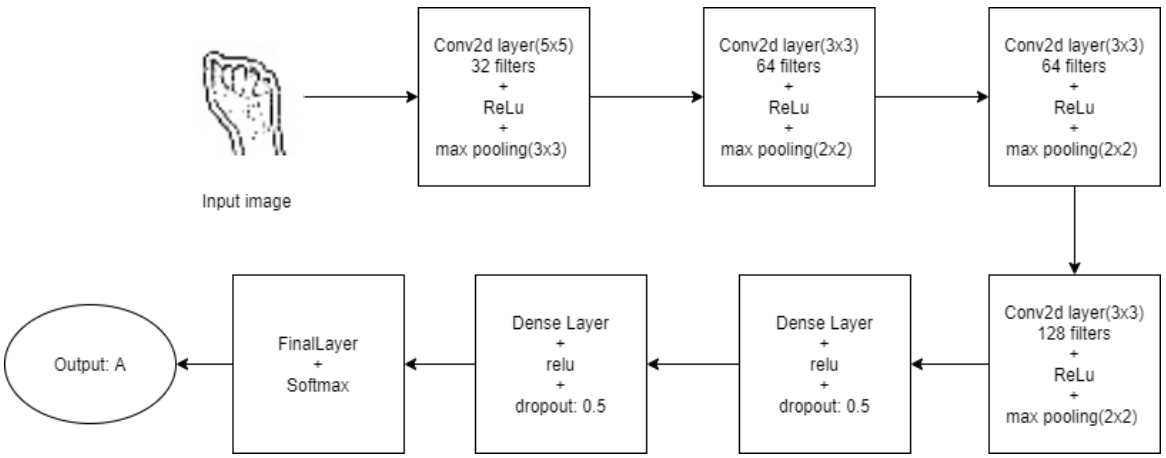

Fig. 3.2.4: CNN Architecture

3.3 Design of the System

Three important steps are followed by the system:

1. Reading the image from the webcam and preprocessing the image.
2. Feeding the preprocessed image to the CNN model.
3. Displaying the predicted text for the input image.

But it also includes some additional features such as skin masking and language translation.

The System works as follows:

1. The system allows the user to select between two options; start(go to 6) and skinmask(go to 2).
2. For skin masking, place your hand on the region of interest and click the mask button.
3. A window with the skin masked output of the video feed is opened.
4. If the hand is properly visible and the background is blacked out, click the save button.
5. If the background is still visible or skin masking is not properly applied then try moving the hand close to the region of interest and keep on pressing “c” until a proper skin masked output is returned(go to 6).
6. An tkinter window with the live feed of the webcam along integreted with the preprocessed roi is visible. It also includes labels to display the recognized text. It also includes an option for selecting language for translation.
7. A dictionary is maintained of the alphabets and 3 custom gestures as keys and counters as the values.
8. A hand gesture is placed in the area of interest and the counters of the frame are calculated and if it is greater than 1000 then the frame is given to the cnn model.
9. The cnn model predicts the image and returns the label.
10. The dictionary values are updated according to the returned label and if any of the counter values in the dictionary is equal to 10 then that letter is displayed on the screen.

Below is the activity diagram of the system:

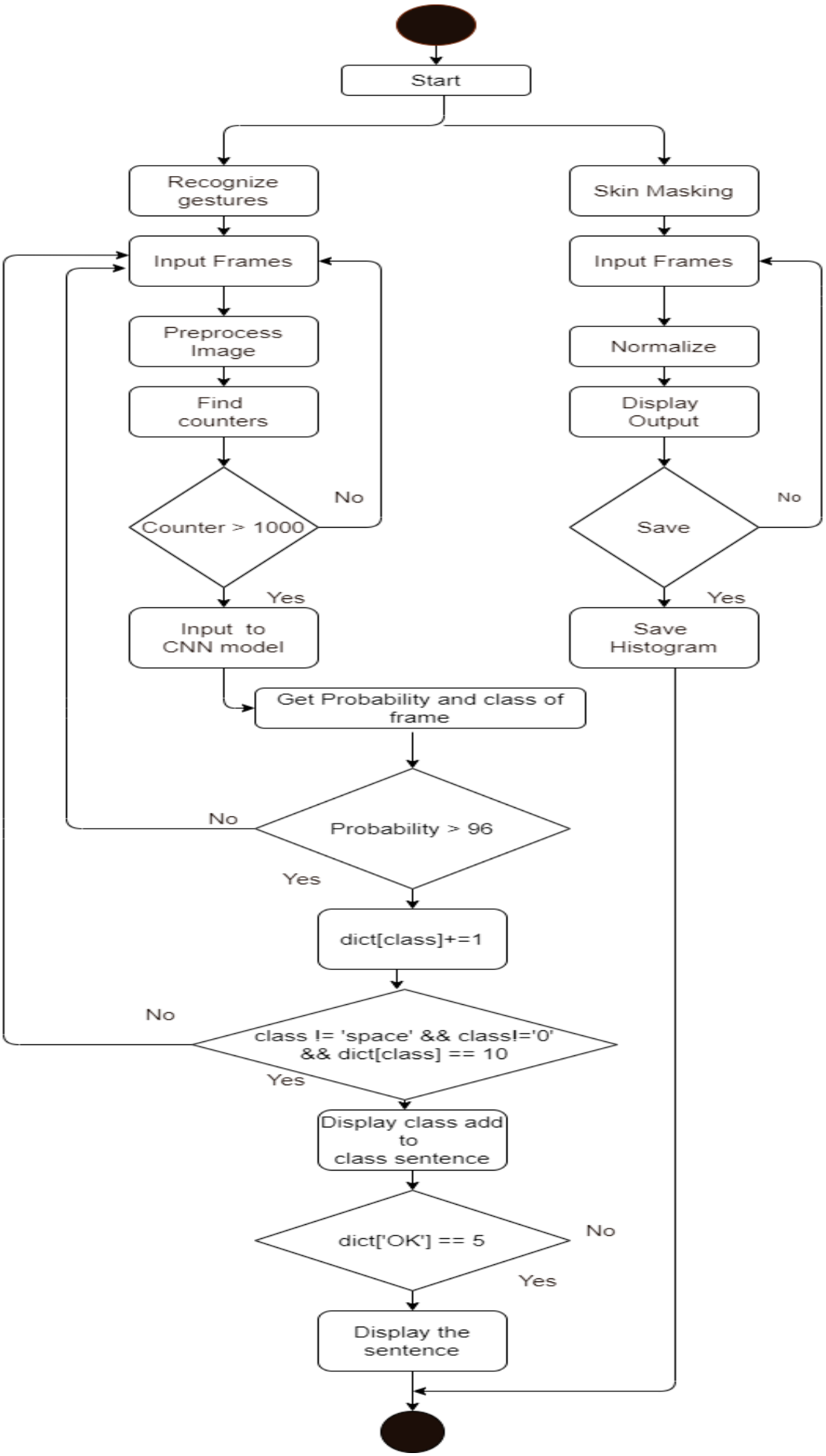

Fig 3.3.1: Activity Diagram

3.4 Hardware/Software Requirement

- Web cam.
- Microsoft Windows 7+ or Linux.
- Python version 3.
- Python Libraries:
    - Sklearn
    - Tensorflow 1.15.0
    - Tensorflow-gpu
    - google_trans_new
    - pyttsx3
    - Open-cv
    - Keras
    - Gtts

3.5 Implementation Details

The system is implemented entirely with python. Tkinter is used to implement the GUI.The input image is preprocessed using one of the many image processing and computer vision libraries supported by python: OpenCV. The CNN model is implemented using tensorflow and keras library.

The steps followed for implementation:

1. Dataset Generation:

    Due to the lack of availability of the raw images of ASL fingerspelling matching the preferred requirement database was created using opencv and python. 6000 images for the 29 classes which includes alphabet a to z, custom gestures for space and full stop were captured using python open cv and webcam. The 6000 images for each class were divided into 3 sets of hands each containing 2000 images to avoid overfitting. Total 174000 images are contained in the dataset. The images are preprocessed while capturing hence the dataset contains preprocessed images.

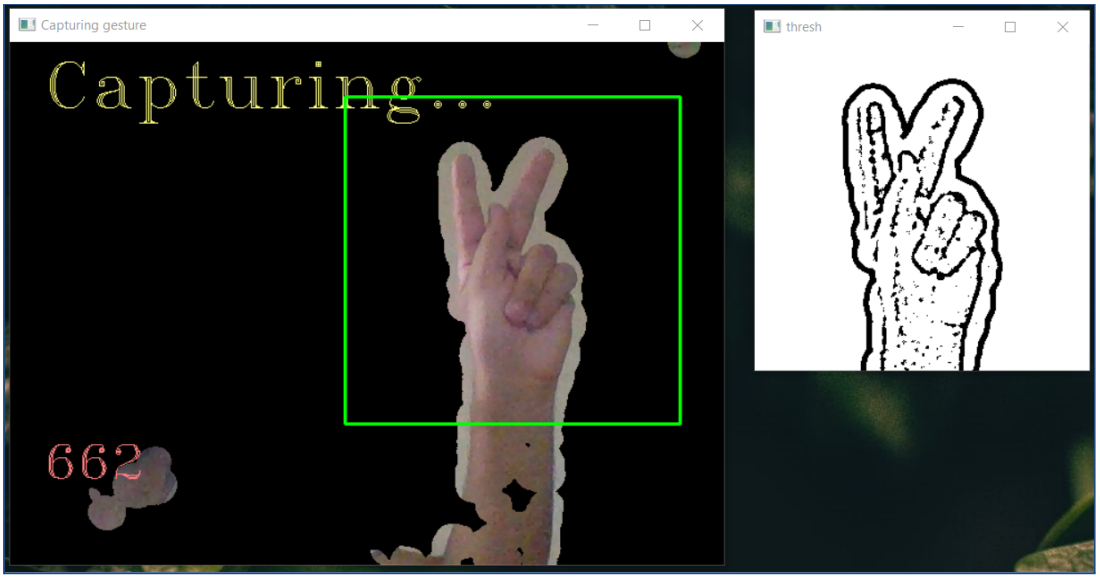

Fig 3.5.1: Capture gestures

1. Create train and test data:

   The dataset created is divided into train and test data. Pickle files for test, train images and train, test label is created. The train pickle files contain 144919.

   Images and the test pickle files contain 14492 images.

2. The gestures present in the dataset:

Fig 3.5.2: Gestures in the dataset

3. Train the CNN model:

The data is trained in convolution neural networks using keras. Convolutional neural networks (ConvNets or CNNs) are more often utilized for classification and computer vision tasks. Prior to CNNs, manual, time-consuming feature extraction methods were used to identify objects in images. However, convolutional neural networks now provide a more scalable approach to image classification and object recognition tasks, leveraging principles from linear algebra, specifically matrix multiplication, to identify patterns within an image. That said, they can be computationally demanding, requiring graphical processing units (GPUs) to train models.

The architecture of the CNN model used to train the dataset:

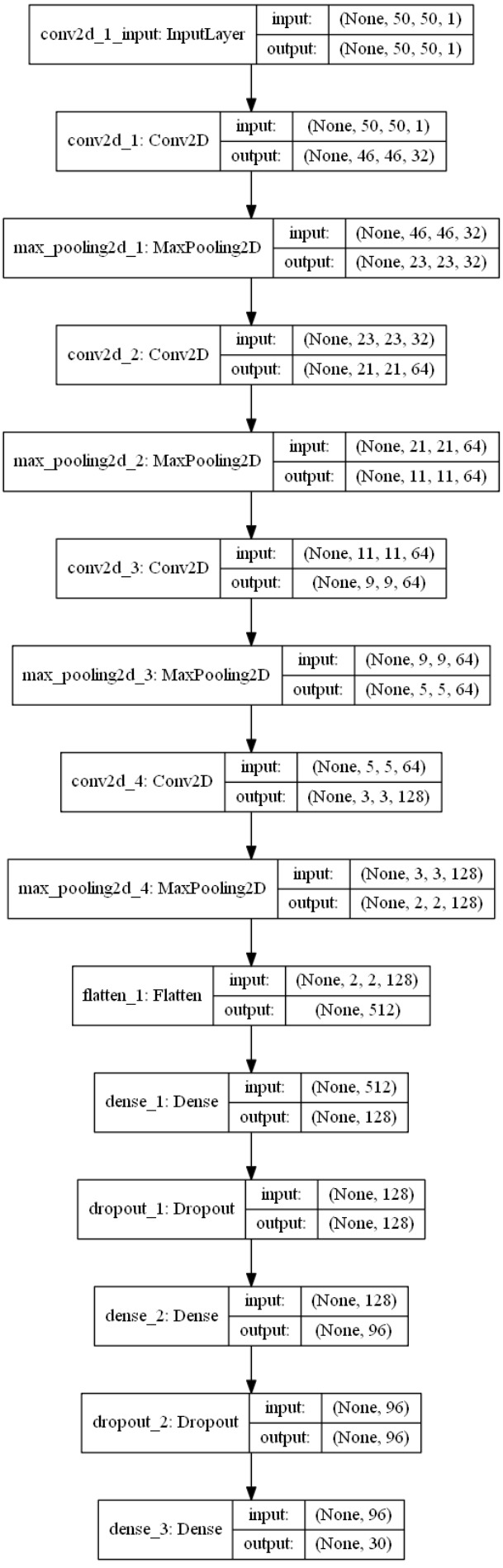

Fig 3.5.3: CNN Model

- Convolution layer: The input image is processed through each convolution layer using the window size of 5x5 for the first 32 bit filter and 3x3 for the rest 64, 64, 128 bits.
- 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.
- ReLu activation Function : We use ReLu (Rectified Linear Unit) in each of the layers. ReLu calculates max(x,0) for each input pixel. This adds nonlinearity to the formula and helps to learn more complicated features. It helps in removing the vanishing gradient problem and speeding up the training by reducing the computation time.
- Dropout Layers: The problem of overfitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn't perform well when given new examples. This layer "drops out" a random set of activations in that layer by setting them to zero. The network should be able to provide the right classification or output for a specific example even if some of the activations are dropped out.
- Softmax activation function: The function is great for classification problems, especially if you're dealing with multi-class classification problems, as it will report back the "confidence score" for each class.

4. Implementing GUI:

The GUI is implemented using Python tkinter.

5. Implementing Sentence formation:

The output section is divided into 4 parts, predicted letter, word, sentence and translated sentence.

If the accuracy returned by the model for the particular frame is greater then 96 only then the predicted class is returned. When a predicted class is returned the counter for that class in the dictionary is incremented by 1. When the counter of the specific class becomes ten that class letter is displayed in the predicted letter section and that letter is appended to the word and the dictionary is cleared. When the custom gesture for space gesture is used the word is appended to the sentence.

6. Implementing text to speech:

   The recognized text and the sentence formed can be given output in the speech form using pyttx library of python. 15

7. Language translator:

   A language translator is implemented using python googletrans library. The translated sentence is converted to speech using the python's playsound library.

Chapter 4

Results and Discussion

4.1 Result and Analysis

The Sign Language Recognition using CNN is implemented as a desktop application. The title window has two options to start the Sign Language Recognition application or to set the skin masking.

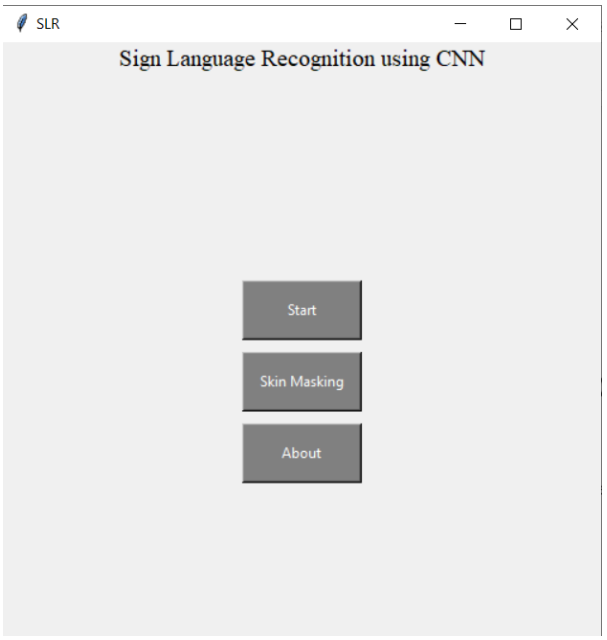

Fig 4.1.1:Front Page

Skin masking allows the user to set the histogram for skin masking according to the features of its skin color.

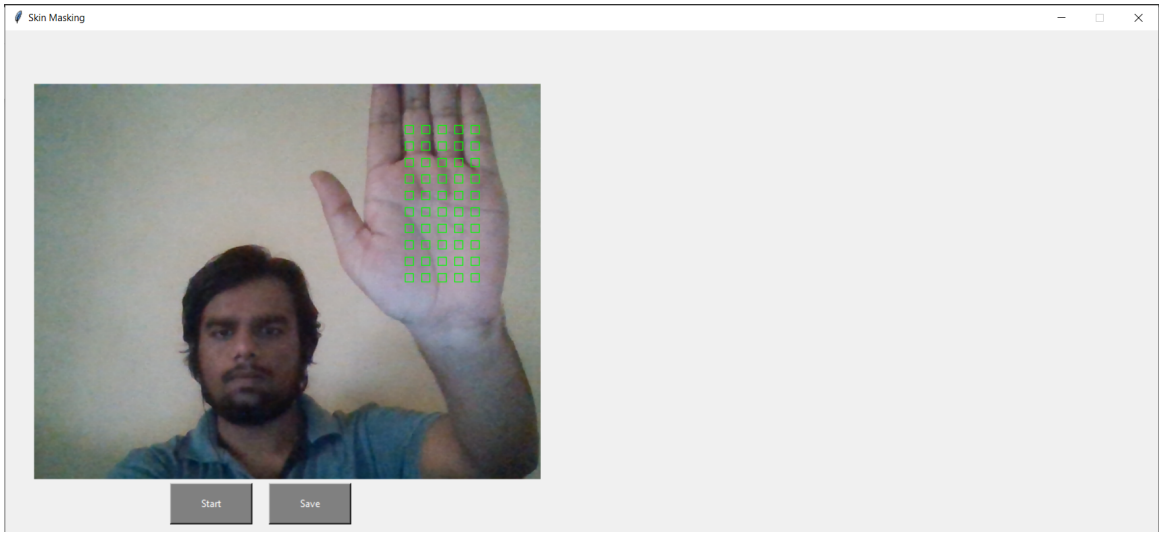

Fig 4.1.2: Skin Masking section(1)

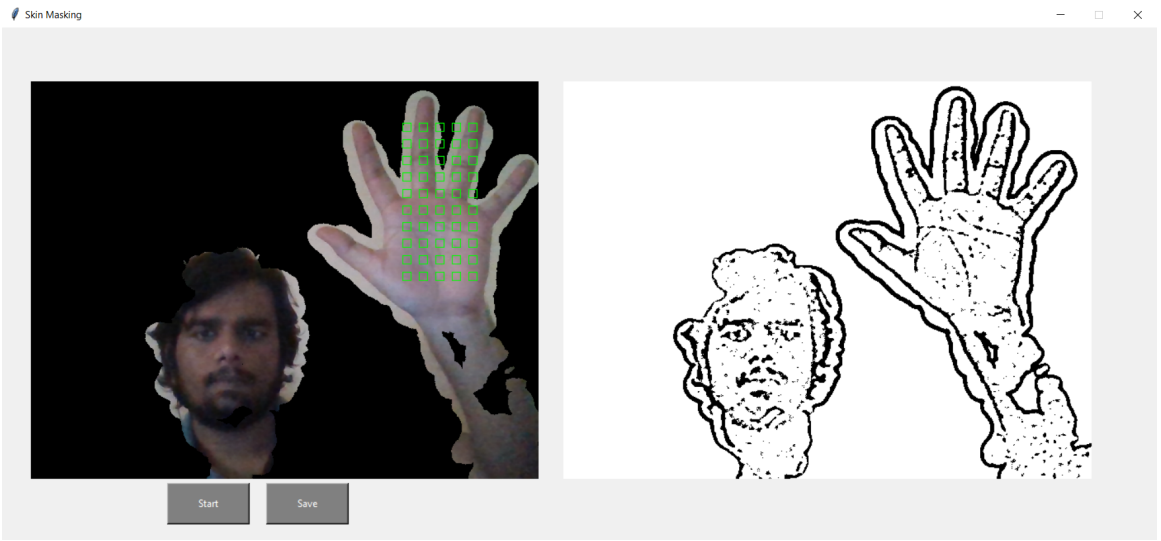

Fig 4.1.3: Skin Masking section(2)

The start button navigates to the sign language recognition section. Which is divided into five parts; live cam feed, recognized text section, word formation and sentence formation section and translated text section.

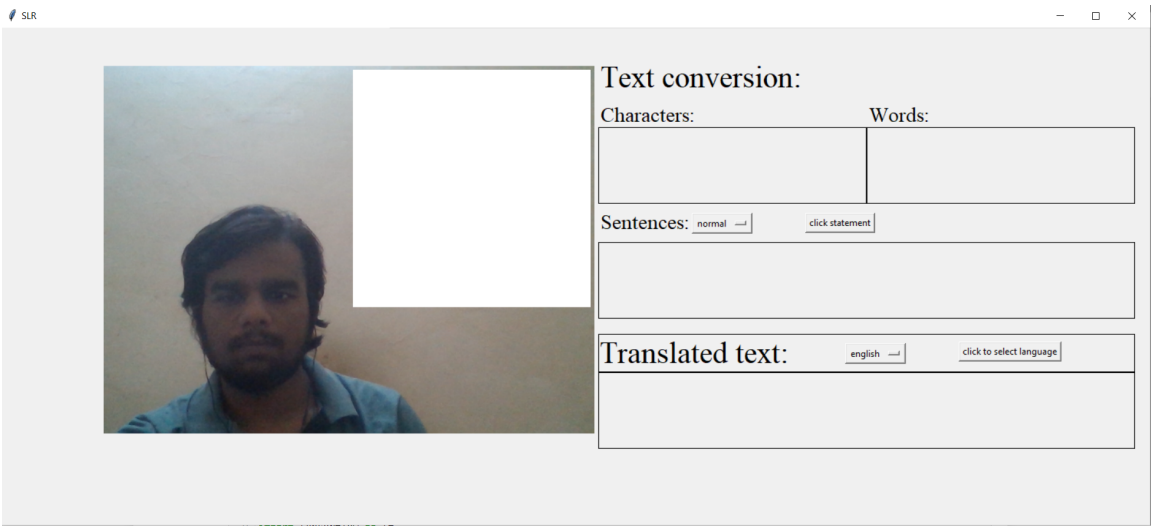

Fig 4.1.4: GUI(1)

The input is taken from the live cam feed and the result is shown in the recognized text section. Two custom gestures are used to form words and a sentence. After the sentence formation is complete the sentence is translated into the specified language. During each letter recognition, sentence formation and translation the output is also given in speech format to that text.

Fig 4.1.5: GUI(2)

The accuracy achieved using Guassian blur threshold using skin masking is 98.79%. Previously only skin masking and thresholding gave the accuracy of 96.55%. So gaussian blur with skin masking is better than thresholding because only thresholding fails to extract different features of the specific hand gesture. The accuracy achieved by gaussian blur thresholding is better than most of the research papers on American sign language recognition. Some research papers also had accuracies above 98% but they involved use of equipment like flex sensors, depth sensors or kinect. The goal of this project was to recognize American sign language fingerspelling with hardware requirements limited to only a computer and a webcam.

Below is the classification report of the testing process:

Classification Report

	precision	recall	f1-score	support
1	0.98	0.99	0.98	495
2	0.98	1.00	0.99	505
3	0.98	1.00	0.99	520
4	0.99	0.99	0.99	506
5	1.00	0.99	0.99	493
6	1.00	1.00	1.00	483
7	0.99	0.97	0.98	512
8	0.97	0.99	0.98	495
9	1.00	1.00	1.00	479
10	1.00	1.00	1.00	487
11	1.00	0.99	0.99	487
12	0.98	0.99	0.98	493
13	0.99	0.99	0.99	521
14	0.99	1.00	0.99	487
15	0.99	0.99	0.99	508
16	0.99	0.99	0.99	527
17	0.99	0.99	0.99	493
18	0.97	0.97	0.97	480
19	0.97	0.96	0.97	519
20	0.98	0.98	0.98	458
21	1.00	1.00	1.00	493
22	0.99	1.00	0.99	496
23	0.98	0.98	0.98	496
24	1.00	1.00	1.00	536
25	1.00	0.98	0.99	501
26	0.98	0.98	0.98	504
27	0.99	0.97	0.98	500
28	0.99	0.99	0.99	518
29	0.97	0.97	0.97	500
accuracy			0.99	14492
macro avg	0.99	0.99	0.99	14492
weighted avg	0.99	0.99	0.99	14492

Fig 4.1.6: Classification Report

Where,

```
Table Name : [('gesture',), ('sqlite_sequence',)]
      g_id   g_name
0        0     test
1        1        a
2        2        b
3        3        c
4        4        d
5        5        e
6        6        f
7        7        g
8        8        h
9        9        i
10      10        k
11      11        l
12      12        m
13      13        n
14      14        o
15      15        p
16      16        q
17      17        r
18      18        s
19      19        t
20      20        u
21      21        v
22      22        w
23      23        x
24      24        y
25      25    space
26      26       ok
27      27        j
28      28        z
29      29   cancel
```

Table 4.1.1: Datasel table

Below is the confusion matrix of our result:

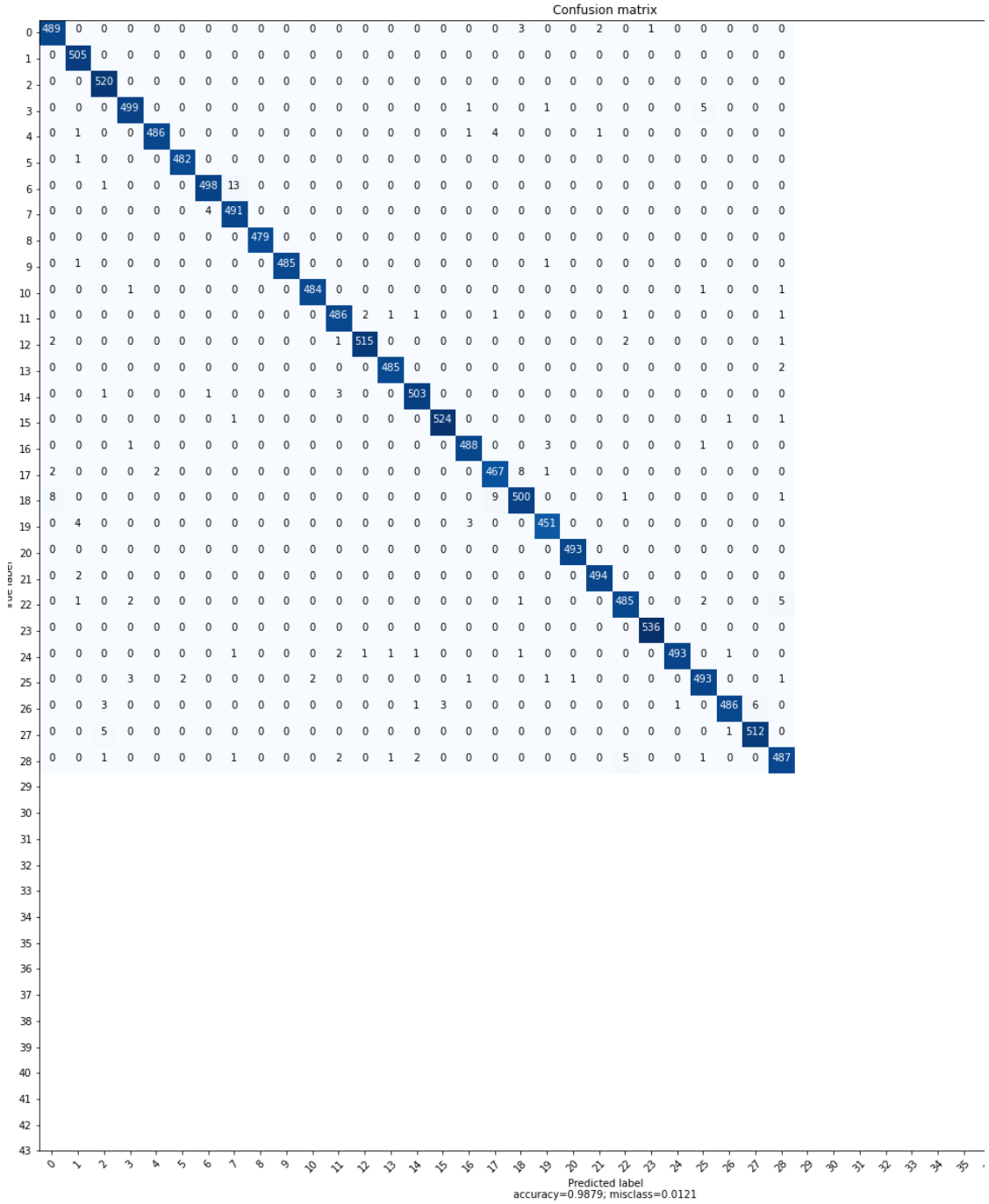

Fig 4.1.7: Confusion Matrix

Chapter 5

Conclusion and Further Work

5.1 Conclusion

Sometimes there is a language barrier between people of different culture, nationality or ethnicity causing language barrier, but for the people with hearing and speech disability there is always a communication gap. Sign Language Recognition bridges the gap by introducing a computer application in the communication path so that the sign language can be automatically captured, recognized and translated to text for the benefit of deaf and mute people. This system uses CNN model which achieved 98.79% accuracy. It also eliminates the problem of background noise. The system pre-processes the image to the required nature for it to be fed into the model. The system is an approach to ease the difficulty in communicating with those having speech disabilities. The amount of training and validation loss observed with the proposed CNN architecture was less.

5.2 Further Work

The proposed system is currently limited to only ASL fingerspelling; thus, it can be enhanced to work with the sign language of different regions like ISL, JSL,etc. The images can be modified by adding pictures with different light density. Further training of the model to achieve efficient detection for two hand gestures. ASL fingerspelling are static gestures but the system can also be modified to capture dynamic gestures as well through word level association of sign language is possible. An android application can be developed which can be used anytime and anywhere.

Bibliography

[1] Sign Language Detection using Image Processing and Deep Learning Teena Varma, Ricketa Baptista, Daksha Chithirai Pandi, Ryland Coutinho, (2018).

[2] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science, (2018).

[3] Lifeprint.com. American Sign Language (ASL) Manual Alphabet (fingerspelling), (2007).

[4] Brandon Garcia, Sigberto Alarcon Viesca "Real-time American Sign Language Recognition with Convolutional Neural Networks", (2017).

[5] https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[6] X. Teng. A hand gesture recognition system based on local linear embedding, April 2005. Journal of Visual Languages and Computing. (2005).

Acknowledgments

We would like to thank our Principal Dr. MD Patil Sir and the Head of Computer Engineering Department, RAIT, Dr. Leena Ragha ma'am for their constant support to all students. We would also like to thank our project coordinator Dr. Dhananjay Dhakane and our project guide Ms. Pallavi Chitte for always guiding and motivating us throughout the course of this project. Last but not the least we would like to thank our friends and families who always stood by us, as an encouraging force to work on this project, even in the midst of a worldwide pandemic.

Date: _____