American Sign Language Recognition

Submitted in partial fulfilment of the requirements

of the degree of

Bachelor of Technology

By

Kapil Mukharu Londhe

(Reg. No.2017BEC002)

Shrikant Santosh Gajewar

(Reg. No.2017BEC006)

Rajat Gangadharrao Gulghane

(Reg. No.2017BEC114)

Supervisor:

Dr. Suhas Gajre

Department of Electronics and Telecommunication Engineering,

Shri Guru Gobind Singhji Institute of Engineering & Technology,

Vishnupuri, Nanded, Maharashtra, India, 431606.

2020-21

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. 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 be cause for disciplinary action 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.

Kapil Mukharu Londhe
(Reg. No.2017BEC002)

Shrikant Santosh Gajewar
(Reg. No.2017BEC006)

Rajat Gangadharrao Gulghane
(Reg. No.2017BEC114)

Date: _____

CERTIFICATE

This is to certify that the report entitled "**American Sign Language recognition**" being submitted by **Kapil Mukharu Londhe** (Reg.No.2017BEC002), **Shrikant Santosh Gajewar** (Reg.No.2017BEC006), **Rajat Gangadharrao Gulghane** (Reg.No.2017BEC114) to **Shri Guru Gobind Singhji Institute of Engineering and Technology, Vishnupuri, Nanded (M.S.), India**, as partial fulfilment for the award of the degree of **Bachelor of Technology** in **Electronics and Telecommunication Engineering**, is a record of bonafide work carried out by him under our supervision and guidance. The matter contained in this report has not been submitted to any other university for the award of any degree or diploma.

Dr. Suhas Gajre
Supervisor
Elect. and Telecom. Engr. Dept.

Dr. A. V. Nandedkar
HOD
Elect. and Telecom. Engr. Dept.

APPROVAL SHEET

This report entitled " American Sign Language recognition " by Kapil Mukharu Londhe, Shrikant Santosh Gajewar, Rajat Gangadharrao Gulghane is approved for the degree of Bachelor of Technology.

Examiners

_____

_____

_____

Supervisor

_____

_____

_____

Date: _____

Place: _____

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to several individuals for supporting us throughout this project. First and foremost, we wish to express our sincere thanks to our supervisor, **Prof. Dr. Suhas Gajre sir**, for his guidance, insightful comments, invaluable suggestions, helpful information, practical advice and unceasing ideas which have always helped us tremendously in this project and writing of this thesis. His immense knowledge, profound experience and professional expertise has enabled us to complete this research successfully.

We also extend gratitude and appreciation to **Dr. A.V. Nandedkar sir** and all the faculties and staff of the Electronics and Telecommunication Department, who have taught us at one point or the other and support towards this project and our team.

We take this opportunity to thank all people who have directly or indirectly helped our project. We pay our respects and gratitude to our teacher and friends for their support and encouragement throughout our project.

Thank you all!

Name of Student

Kapil Mukharu Londhe

(Reg. No.2017BEC002)

Shrikant Santosh Gajewar

(Reg. No.2017BEC006)

Rajat Gangadharrao Gulghane

(Reg. No.2017BEC114)

ABSTRACT

Sign language is one of the oldest and most natural form of language for communication, but since most people do not know sign language and interpreters are very difficult to come by, we have come up with a real time method using neural networks for fingerspelling based American Sign Language. In our method, the hand is first passed through a filter and after the filter is applied the hand is passed through a classifier which predicts the class of the hand gestures. Our method provides 98.23 % accuracy for the 26 letters of the alphabet.

Keywords: Convolutional Neural Networks (CNN), TensorFlow, Keras, OpenCV, Classification

CONTENTS

List of Figures	**i**
List of Tables	**ii**
Abbreviations, Notations and Nomenclature	**iii**
1. Introduction	**1**
2. Literature Survey	**3**
3. Pre -Requirements of Projects	**4**
3.1 Data Collection	**4**
3.2 Data Pre-processing	**4**
4. Proposed Method	**5**
4.1 Data Set Generation	**5**
4.2 Image classification model	**5**
5. Experimental Results	**9**
5.1 Training and Validation	**9**
5.2 Testing	**10**
6. Conclusions and Future Scope	**11**
6.1 Conclusion	**11**
6.2 Future work	**11**
7. References	**12**

List of Figures

Figure	Page no.
1) Hand signs for American sign language alphabets	2
2) Examples of American sign language alphabets in dataset.	4
3) Fig. Pre-processed images from dataset.	4
4) CNN Model	7
5) Methodology Flowchart	8
6) Epoch vs Model Accuracy and Epoch vs Model Loss	9
7) ASL alphabet “A” recognised in real time.	10
8) ASL alphabet “B” recognised in real time.	10

List of Tables

Table	Page no.
1) Sign Language components	1

Abbreviations, Notations and Nomenclature

1) ASL: American Sign Language
2) ANN: Artificial Neural Network
3) CNN: Convolutional Neural Network
4) OpenCV: Open Computer Vision
5) RGB: Red, Green and Blue
6) RMSprop: Root Mean Square Propagation

CHAPTER 1
INTRODUCTION

A real-time sign language translator is an important milestone in facilitating communication between the deaf community and the general public. The goal of this project was to build a neural network able to classify which letter of the American Sign Language (ASL) alphabet is being signed, given an image of a signing hand. This project is a first step towards building a possible sign language translator, which can take communications in sign language and translate them into written and oral language. Such a translator would greatly lower the barrier for many deaf and mute individuals to be able to better communicate with others in day-to-day interactions. This goal is further motivated by the isolation that is felt within the deaf community. Loneliness and depression exist in higher rates among the deaf population, especially when they are immersed in a hearing world. Large barriers that profoundly affect life quality stem from the communication disconnect between the deaf and the hearing. The objective of this project was to see if neural networks can classify signed ASL letters using simple images of hands taken with a personal device such as a laptop webcam. This is in alignment with the motivation as this would make a future implementation of a real time ASL-to-oral/written language translator practical in an everyday situation.

Sign language is a visual language and consists of 3 major components:

Fingerspelling	Word level sign vocabulary	Non-manual features
Used to spell words letter by letter .	Used for the majority of communication.	Facial expressions and tongue, mouth and body position.

Fig 1.1. Sign Language

ASL possesses a set of 26 signs known as the American manual alphabet, which can be used to spell out from the English language. Such signs make use of the 19 handshapes of American Sign Language.

In our project we basically focus on producing a model which can recognise Fingerspelling based hand gestures in order to form a complete word by combining each gesture. The gestures we aim to train are as given in the image below.

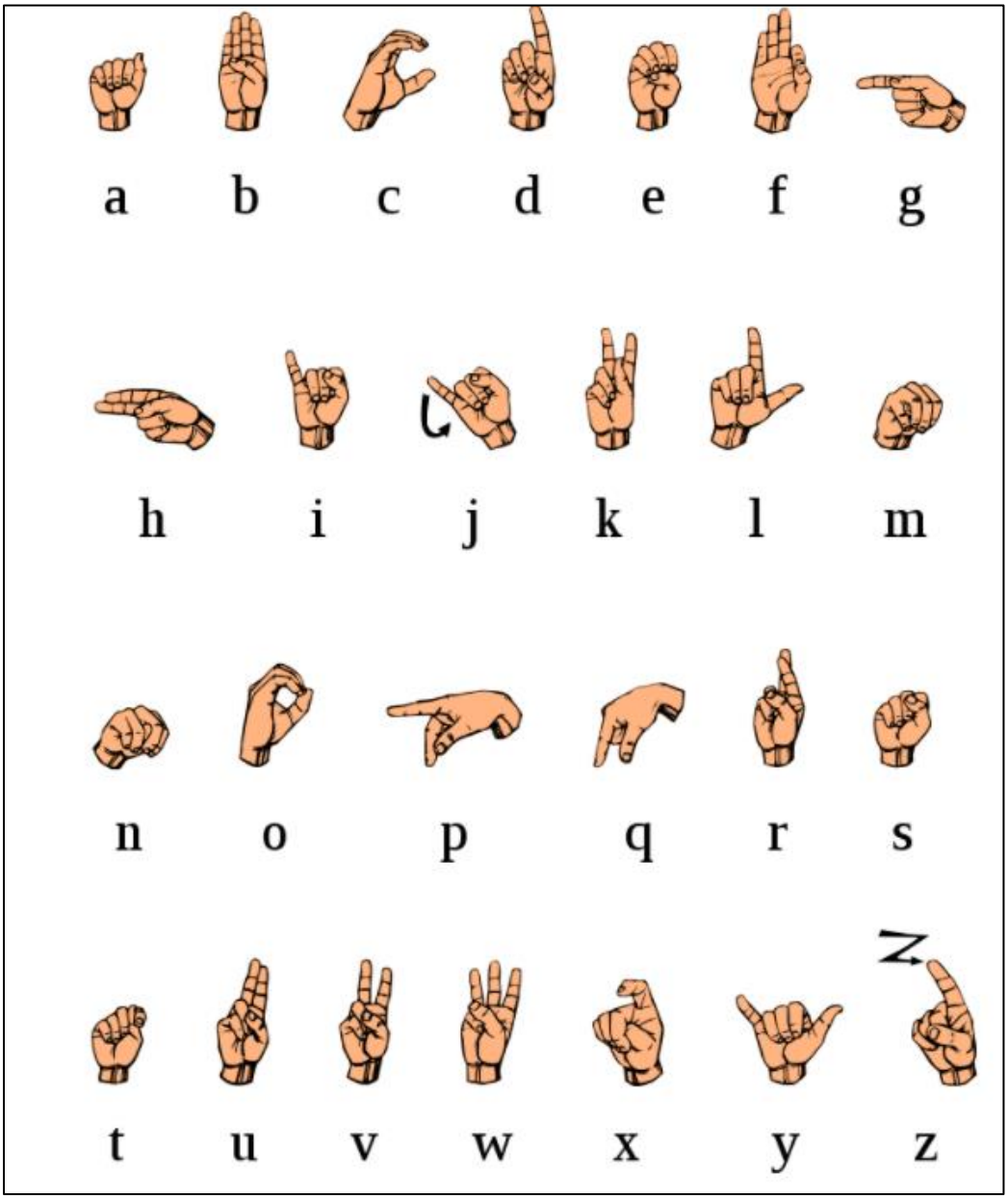

Fig 1.2. Hand signs for American sign language alphabets

CHAPTER 2

LITERATURE SURVEY

Literature review of our proposed system shows that there have been many explorations done to tackle the sign recognition in videos and images using several methods and algorithms.

“He Siming [1] proposed a system having a dataset of 40 common words and 10,000 sign language images. To locate the hand regions in the video frame, Faster R-CNN with an embedded RPN module is used. It improves performance in terms of accuracy. Detection and template classification can be done at a higher speed as compared to single stage target detection algorithm such as YOLO. The detection accuracy of Faster R-CNN in the paper increases from 89.0% to 91.7% as compared to Fast-RCNN.

A similar work was done by J Huang [2] He created his own dataset using Kinect and got a total a total of 25 vocabularies which are used in everyday lives. He then applied a 3D CNN in which all kernels are also in 3D. The input of his model consisted of 5 important channels which are colour-r, colour-b, colour-g, depth and body skeleton. He got an average accuracy of 94.2%

In one of the research done by M. Geetha and U. C. Manjusha [3], They used 50 specimens of every alphabets and digits in a vision based recognition of Indian Sign Language characters and numerals using B-Spine approximations. The region of interest of the sign gesture is analysed and the boundary is removed. The boundary obtained is further transformed to a B-spline curve by using the Maximum Curvature Points (MCPs) as the Control points. The B-spline curve undergoes a series of smoothening process so features can be extracted. Support vector machine is used to classify the images and the accuracy is 90.00%.

In [4], a low-cost approach has been used for image processing. The capture of images was done with a green background so that during processing, the green colour can be easily subtracted from the RGB colour space and the image gets converted to black and white. The sign gestures were in Sinhala language. The method that thy have proposed in the study is to map the signs using centroid method. It can map the input gesture with a database irrespective of the hands size and position. The prototype has correctly recognised 92% of the sign gestures.

NOTE: [1] denotes the reference number

CHAPTER 3

PRE-REQUIREMENTS OF PROJECT

3.1 Data Collection

The dataset is comprised of 166,000 images which are 400x400 pixels. There are 28 total classes, each with 5996 images, 26 for the letters A-Z and 2 for space and nothing. This data is created with the images taken from his laptop's webcam. These photos were then cropped, rescaled, and labelled for use. The complete dataset is uploaded on Kaggle [5].

1) ASL sign "A"

2) ASL sign "B"

3) ASL sign "C"

Fig 3.1. Examples of American sign language alphabets in dataset.

3.2 Data Pre-processing

Data pre-processing is done using OpenCV library of Python. The dataset consists of colour images, these images are first converted into grayscale images and then resized into size (150x150).

1) ASL sign "A"

2) ASL sign "B"

3) ASL sign "C"

Fig 3.2. Pre-processed images from dataset.

CHAPTER 4

PROPOSED METHOD

4.1 Data Set Generation

We used Open computer vision (OpenCV) library in order to produce our dataset. We captured around 5996 images of each of the symbol in ASL for training purposes and around 4 images per symbol for testing purpose.

4.2 Image classification model

Our ASL letter classification is done using a convolutional neural network (CNN or ConvNet). CNNs are machine learning algorithms that have seen incredible success in handling a variety of tasks related to processing videos and images. Since 2012, the field has experienced an explosion of growth and applications in image classification, object localization, and object detection. A primary advantage of utilizing such techniques stems from CNNs abilities to learn features as well as the weights corresponding to each feature. Like other machine learning algorithms, CNNs seek to optimize some objective function, specifically the loss function.

Convolutional Neural Network (CNN) Model:

1. 1st Convolution Layer: The input picture has resolution of 150x150 pixels. It is first processed in the first convolutional layer using 64 filter weights (7 x 7 pixels each). This will result in a 144x144x64 pixel image, one for each Filter-weights.
2. 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 72x72x64 pixels.
3. 2nd Convolution Layer: Now, these 72x72x64 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 64 filter weights (7x7 pixels each). This will result in a 33x33x64 pixel image.
4. 1st Dropout Layer: A dropout of 0.5 is applied to the output of previous layer.
5. 3rd Convolution Layer: Now, these 33x33x64 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 256 filter weights (3x3 pixels each). This will result in a 31x31x256 pixel image.
6. 2nd 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 15x15x256 pixels.
7. 4th Convolution Layer: Now, these 15x15x256 from the output of the second pooling layer is served as an input to the fourth convolutional layer. It is processed in the second convolutional layer using 256 filter weights (3x3 pixels each). This will result in a 7x7x256 pixel image.

8. 3rd 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 3x3x256 pixels.
9. 1st Densely Connected Layer: Now these images are used as an input to a fully connected layer with 512 neurons and the output from the second convolutional layer is reshaped to an array of 3x3x256 =2304 values. The input to this layer is an array of 2304 values. The output of these layer is fed to the final Layer. We are using a dropout layer of value 0.5 to avoid overfitting.
10. Final layer: The output of the 1st 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 i.e., 28 classes (26 alphabets + space + nothing)

Activation Function: We have used ReLU (Rectified Linear Unit) in each of the layers (convolutional as well as fully connected neurons). 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.

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.

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 does not 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.

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).

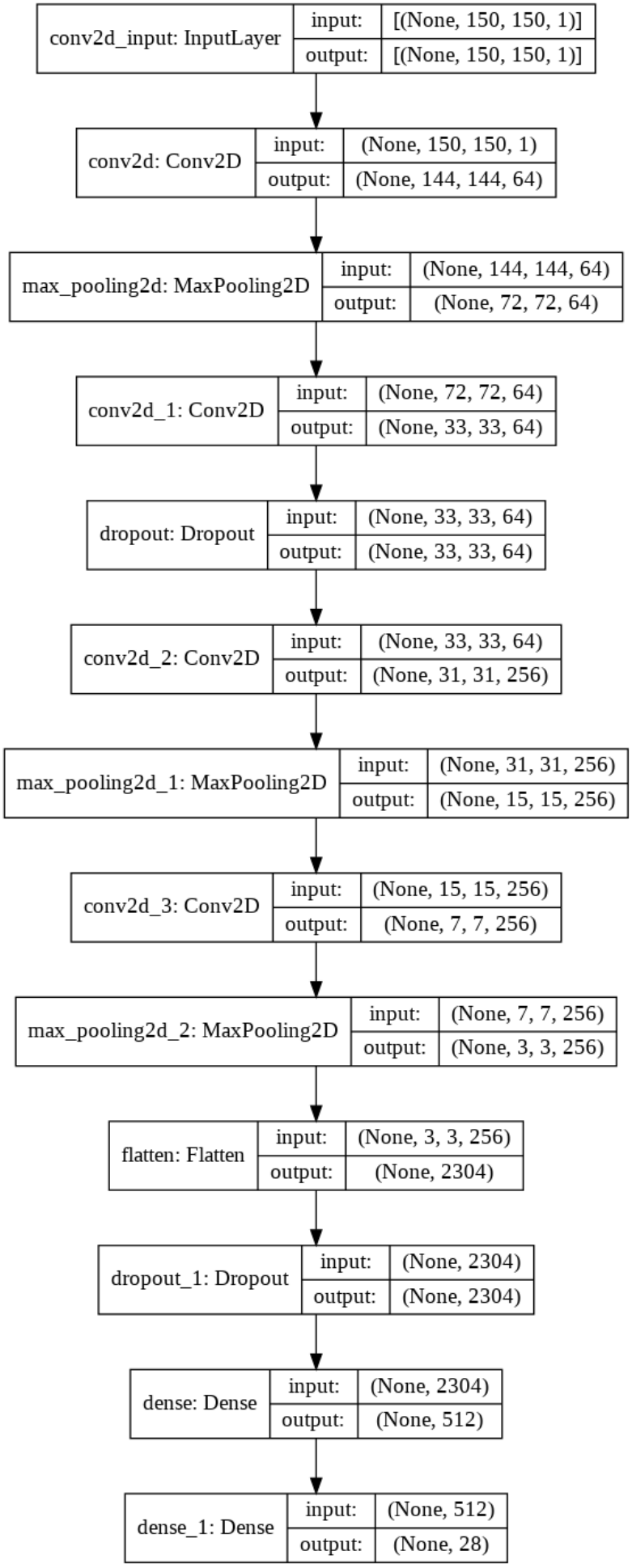

Fig 4.1. CNN Model

Methodology Flowchart:

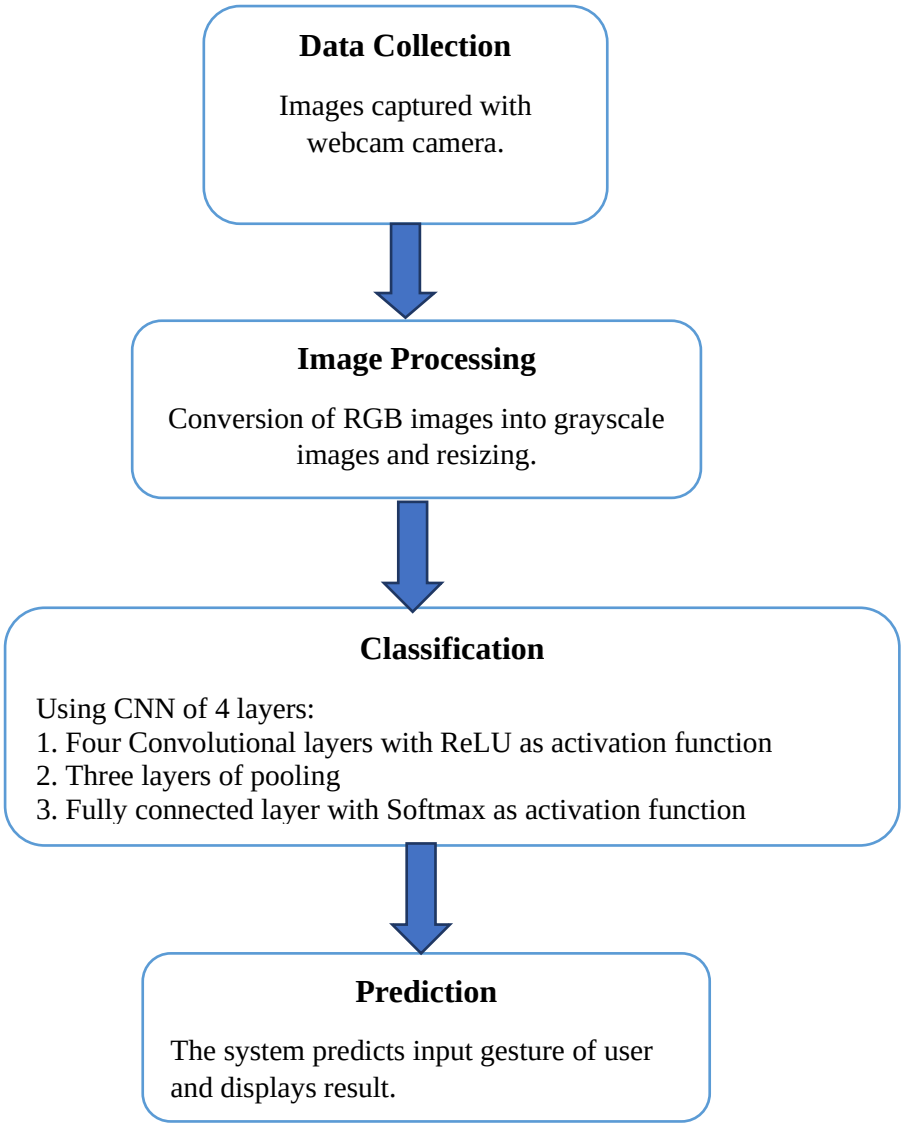

Fig 4.2. Methodology Flowchart

CHAPTER 5

Experimental Results

5.1 Training and Validation

Our models was trained using Adam optimizer and Cross Entropy Loss. Adam optimizer is known for converging quickly in comparison with Stochastic Gradient Descent (SGD). Training is done with the batch size of 256 and for 10 Epochs. The total time taken for training is around 98 minutes. The peak validation accuracy achieved is 98.23%.

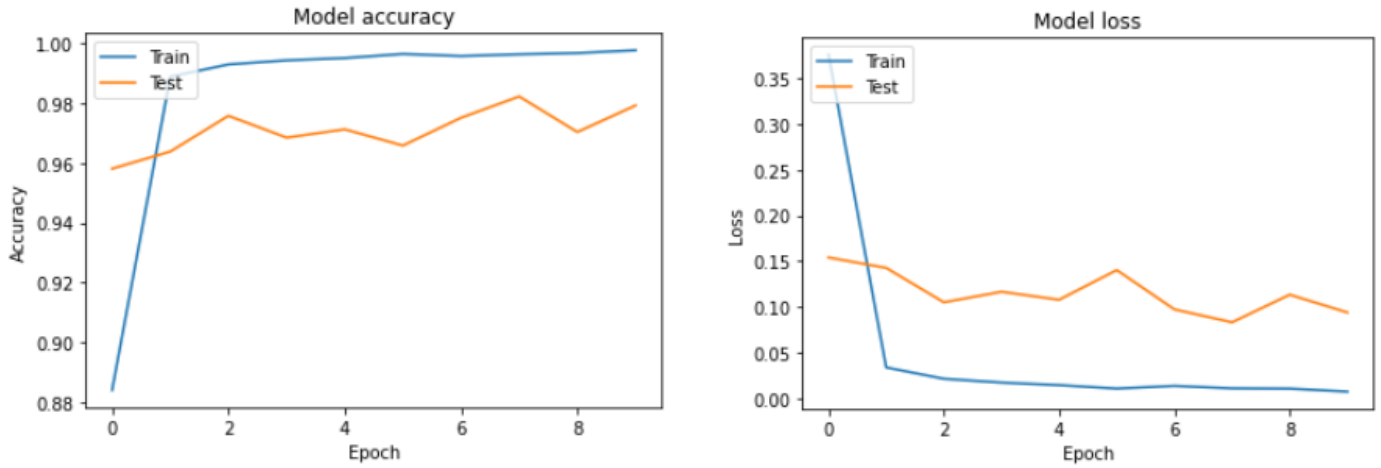

Fig 5.1. Epoch vs Model Accuracy and Epoch vs Model Loss

5.2 Testing

The classification model is tested real time on hand signs gives excellent results. Some of the results are shown below:

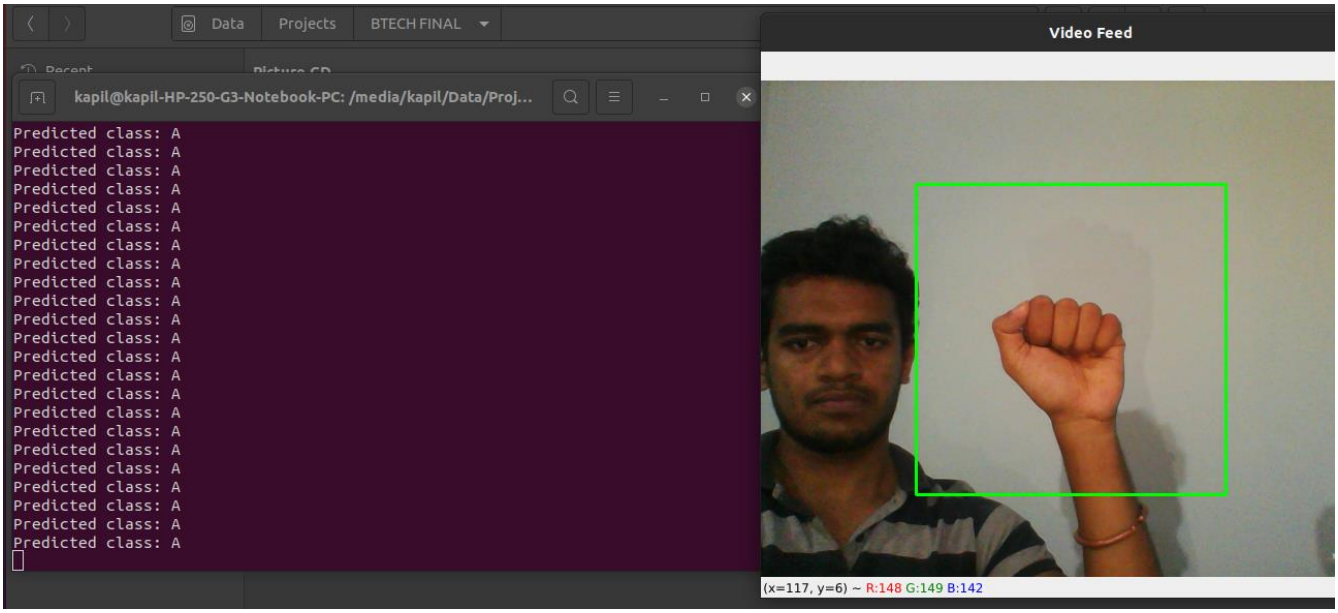

Fig 5.2. ASL alphabet “A” recognised in real time.

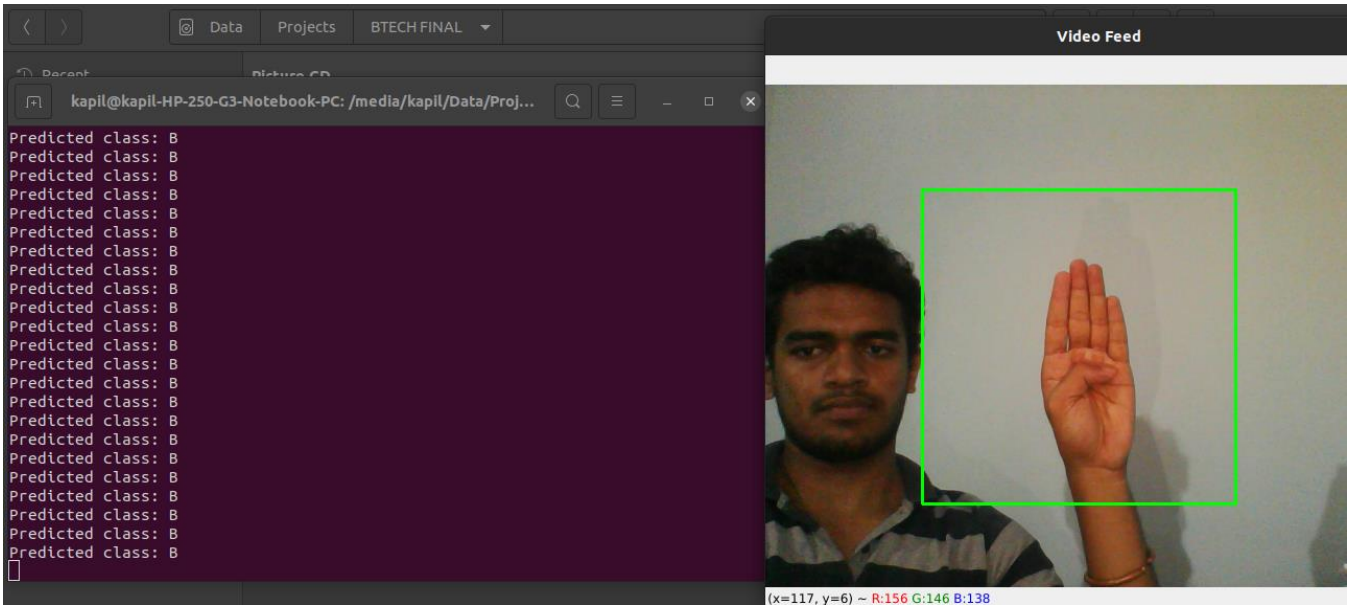

Fig 5.3. ASL alphabet “B” recognised in real time.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this project, we have implemented an automatic sign language gesture recognition system in real-time, using Convolutional Neural Network. We learned about how sometimes basic approaches work better than complicated approaches. We also realized the time constraints and difficulties of creating a dataset from scratch. Looking back, it would have been nice to have had a dataset already to work off. Some letters were harder to classify in our live demo such as “I” vs “J” since they only differ by a very small edge (hand orientation is different). Although our classification system works quite well and gained an accuracy of 98.23% as has been demonstrated through tables and images, there is still a lot of scope for possible future work.

6.2 Future Wok

Possible extensions to this project would be extending the gesture recognition system to all alphabets of the ASL and other non-alphabet gestures as well. Having used GOOGLE COLAB as the platform for implementation, we feel that we can also improve upon the speed of our real-time system by using some advanced platforms. The framework of this project can also be extended to several other applications like controlling robot navigation using hand gestures, enable the deaf and hard-of-hearing equal access to video consultations, whether in a professional context or while trying to communicate with their healthcare providers via telehealth, instead of using basic chat, these advancements would allow the hearing-impaired access to effective video communication.

We look forward to using more things in our datasets and improve the model so that it recognises more analytical communication features while at the same time get a high accuracy. We would also like to enhance the system by adding speech recognition so that blind people can benefit as well.

References

[1] He, Siming. (2019). Research of a Sign Language Translation System Based on Deep Learning. 392-396.10.1109/AIAM48774.2019.00083.

[2] Huang, J., Zhou, W., & Li, H. (2015). Sign Language Recognition using 3D convolutional neural networks. IEEE International Conference on Multimedia and Expo (ICME) (pp. 1-6). Turin: IEEE.

[3] M. Geetha and U. C. Manjusha, "A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation", International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 3, pp. 406-415. 2012.

[4] Herath, H.C.M. & W.A.L.V.Kumari, & Senevirathne, W.A.P.B & Dissanayake, Maheshi. (2013). IMAGE BASED SIGN LANGUAGE RECOGNITION SYSTEM FOR SINHALA SIGN LANGUAGE

[5] American Sign Language Dataset https://www.kaggle.com/kapillondhe/american-sign-language