HAND SIGN DETECTION USING DEPTH SENSOR CAMERA

A Thesis

Submitted in partial fulfillment of the requirements

for the award of the Degree of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

By

S. APURVA (16011A0403)

MD.YASER ALI TARIQ (16011A0421)

G. PRABHU NITHIN (16011A0431)

D.SUNANDA (16011A0456)

Under the esteemed Supervision of

Dr. K. ANITHA SHEELA

Professor and Head of the Department

Department of ELECTRONICS AND COMMUNICATION ENGINEERING

JNTUHCEH

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

JNTUH COLLEGE OF ENGINEERING HYDERABAD-500085

TELANGANA (2019-2020)

JNTUH COLLEGE OF ENGINEERING

HYDERABAD-500085

ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled as "**HAND SIGN DETECTION USING DEPTH SENSOR CAMERA**" a bona fide record of the work done by **S.APURVA, MD.YASER ALI TARIQ, G.PRABHU NITHIN, D.SUNANDA** bearing Admission no's: 16011A0403, 16011A0421, 16011A0431, 16011A0456 submitted to the faculty of Electronics and Communication Engineering in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY with specialization in **ELECTRONICS AND COMMUNICATION ENGINEERING** from Jawaharlal Nehru technological University Hyderabad, College of Engineering (Autonomous), Hyderabad during the year **2019-2020.**

Signature of the Supervisor

Dr. K. ANITHA SHEELA

Professor and Head of the Department

Department of Electronics and Communication Engineering, JNTUH.

JNTUH COLLEGE OF ENGINEERING HYDERABAD-500085

ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled as "**HAND SIGN DETECTION USING DEPTH SENSOR CAMERA**" a bona fide record of the work done by **S.APURVA, MD.YASER ALI TARIQ, G.PRABHU NITHIN, D.SUNANDA** bearing Admission no's: 16011A0403, 16011A0421, 16011A0431, 16011A0456 submitted to the faculty of Electronics and Communication Engineering in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY with specialization in **ELECTRONICS AND COMMUNICATION ENGINEERING** from Jawaharlal Nehru technological University Hyderabad, College of Engineering (Autonomous), Hyderabad during the year **2019-2020.**

Signature of the Head of the Department

Dr. K. ANITHA SHEELA

Professor and Head of the Department

Department of Electronics and Communication Engineering, JNTUH.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible.

We would like to take this opportunity to thank all those who helped us in the successful completion of this project.

We would like to express our deep and sincere thanks to Dr. K. Anitha Sheela, Professor and Head of the Department of ECE, JNTUHCEH, Hyderabad for her constant support and encouragement.

We are greatly indebted to our guide Dr. K ANITHA SHEELA, HOD & Professor of department of ECE, JNTUHCEH, for her valuable suggestions and guidance in our endeavor. Without her willing disposition, spirit of accommodation, timely clarification and above all faith in us, this project could not have been completed in due time.

-S. Apurva

-Md. Yaser Ali Tariq

-G. Prabhu Nithin

-D. Sunanda

TABLE OF CONTENTS

GLOSSARY

ASL – American Sign Language

Depth – distance calculated from center of IR camera

Depth Map – Depth image constructed from depth values

HCI – Human Computer Interaction

ISL – Indian Sign Language

Processing – Java based graphical library

RGB – Red, Green, Blue

Tensor – generalization of scalars, vectors and matrices

LIST OF FIGURES

ABSTRACT

Inability to speak is considered to be true disability. People with this disability use different modes to communicate with others, there are a number of methods available for their communication one such common method of communication is sign language

Developing sign language application for deaf people can be very important, they'll be able to communicate easily with even those who don't understand sign language. Our project aims at taking the basic step in bridging the communication gap between normal people, deaf and dumb people using sign language.

The main focus of this work is to create a vision-based system to identify sign language gestures from the video sequences. The reason for choosing a system based on vision relates to the fact that it provides a simpler and more intuitive way of communication between a human and a computer. In this report, 10 different gestures have been considered

We used the following approach for the classification of sign language Gestures:

- Images contain both the temporal as well as the spatial features. So we have used two different models to train both the RGB as well as the spatial features.
- A combined model is also trained by mapping RGB and Depth information into a 4d tensor.
- To train the model on the spatial features we have used VGG 16 model which is a deep CNN (convolutional neural net) CNN was trained on individual RGB and DEPTH train data.
- Trained CNN model was used to make predictions for individual frames to obtain a sequence of predictions or pool layer outputs for each class.
- The data set used consists of Indian Sign Language (ISL) Gestures, with around 150 images belonging to 10 gestures categories. Using the predictions by CNN 94% accuracy was obtained on train data and 86% accuracy on validation data.

Keywords: Depth Camera, RGB-D, CNN, ASL, ISL.

CHAPTER 1: INTRODUCTION & PRELIMINARIES

1.1 Introduction to Sign Language:

Deaf people around the world communicate using sign language as distinct from spoken language in their everyday a visual language that uses a system of manual, facial and body movements as the means of communication.

Sign language is not a universal language, and different sign languages are used in different countries, like the many spoken languages all over the world. Hundreds of sign languages are in used around the world, for instance, Japanese Sign Language, British Sign Language (BSL), Spanish Sign Language, Turkish Sign Language etc.

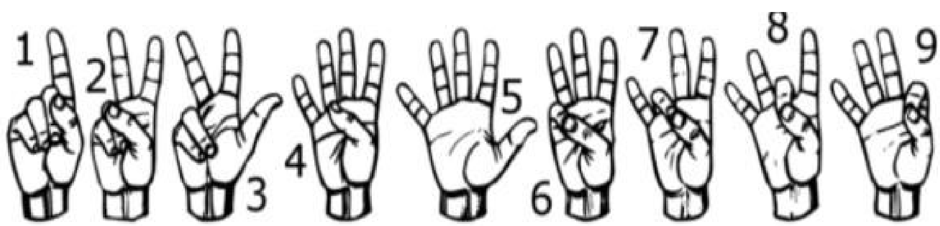

Fig.1.1 Hand Signs for numbers

1.2 About Sign Language:

Motion of any body part like hand is a form of gesture. Here for gesture recognition we are using image processing, and computer vision Gesture recognition enables computer to understand human actions and also acts as an interpreter between computer and human. This could provide potential to human to interact naturally with the computers without a physical contact of the mechanical devices.

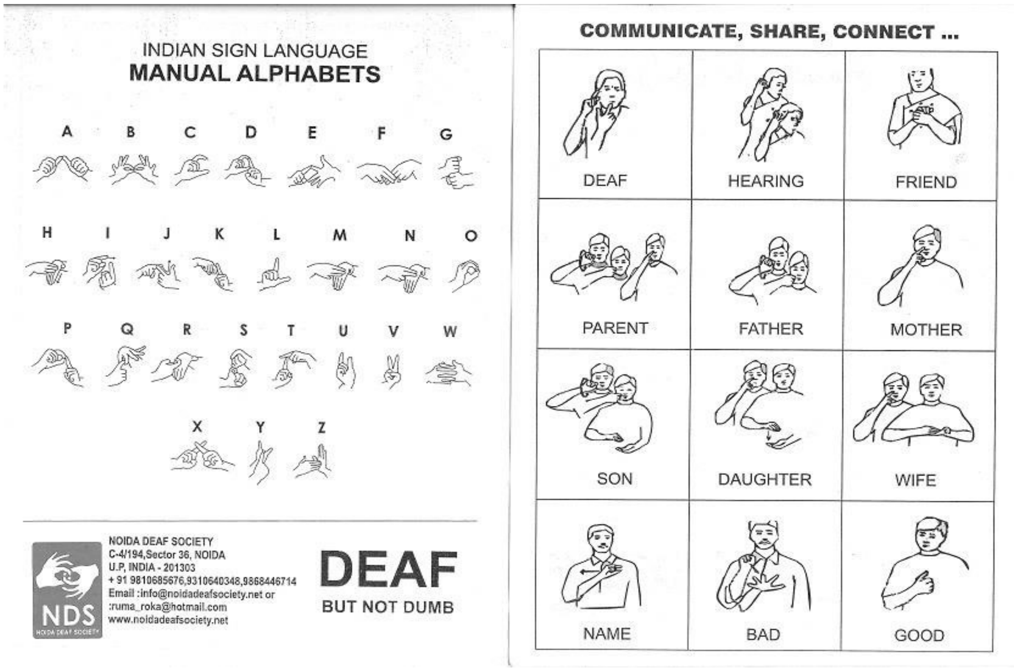

Fig.1.2 Alphabets and basic words in Sign Language

Gestures are performed by deaf and dumb community to perform sign language. This community used sign language for their communication when broadcasting audio is impossible, or typing and writing is difficult, but there is the vision possibility. At that time sign language is the only way for exchanging information between people.

Normally sign language is used by everyone when they do not want to speak, but this is the only way of communication for deaf and dumb community Sign language is also serving the same meaning as spoken language does. This is used by deaf and dumb community all over the world but in their regional form like ISL, ASL.

Sign language can be performed by using Hand gesture either by one hand or two hands. It is of two types Isolated sign language and continuous sign language Isolated sign language consists of single gesture having single word while continuous ISL or Continuous Sign language is a sequence of gestures that generate a meaningful sentence. In this report we performed isolated ASL gesture recognition technique using RGB images, Depth Images and a combined technique.

1.3 Need for Sign Language recognition:

- Sign Language recognition has emerged as one of the most important research areas in the field of human computer interaction (HCI).
- The Sign Language Recognition System (SLR) is highly desired due to its ability to overcome the barrier between deaf and hearing people.
- Sign languages can be used when the spoken word is physically impossible, such as talking underwater, talking through glass, from a distance, at a loud music concert, and talking with your mouth full.

CHAPTER 2: COMPONENTS AND SOFTWARES

2.1 Depth Sensor Camera (Prime Sense by ASUS):

Primesense Carmine 1.09 Short-range 3D Camera Sensor is for shortrange scanning. It can capture motion just like the Kinect scanner which is based on the Primesense technology.

The Primesense 3D Depth Sensor is an end-to-end solution that enables devices to perceive the world in 3D and to translate these perceptions into a synchronized image, in the same way that people do, thus harnessing the power of Natural Interaction. The solution includes a sensor component, which comprehends the user's interaction within these surroundings.

The 3D depth Sensor sees and tracks user movements within a scene. All sensor activity is performed without any assumptions about the user or the environment. The sensor includes several optional sensory input capabilities: Depth (3D) Image, color (RGB) image and audio (the sensor has 2 microphones).

All sensory information (depth image, color image and audio) is transferred to the host via a USB2.0 interface with complete timing alignment.

Fig 2.1 Depth Sensor camera

2.2 Camera Specifications:

Field of View: 54 degrees horizontal, 45 degrees vertical
Depth Image Size: VGA (640x480)
Spatial X/Y Resolution@0.5m: 1mm
Depth Z Resolution @0.5m: <1mm
Maximum Image Throughput: (QVGA) 60fps, (VGA) 30fps
Operation Range: 0.35-3m
Color Image Size: 1280x960
Audio: Built-in Microphones-two mics
Data Interface: USB 2.0, USB 3.0
Power Consumption: 2.25W
Power Supply: USB 2.0, USB 3.0
Dimension: 18x2.5x3.5cm
Operation Environment: Indoor

2.3 Depth sensing technologies:

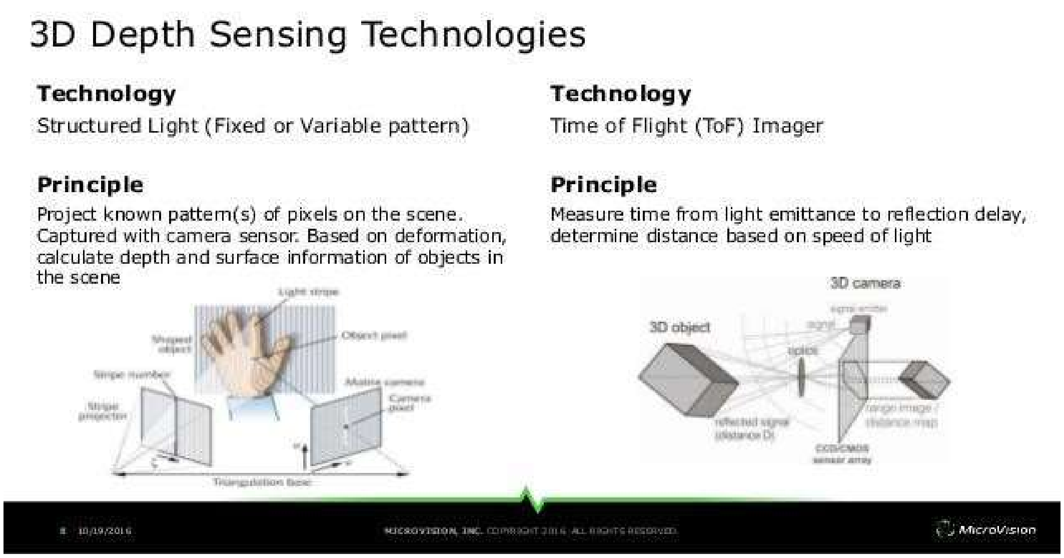

Fig 2.2.a Depth Sensing Technology

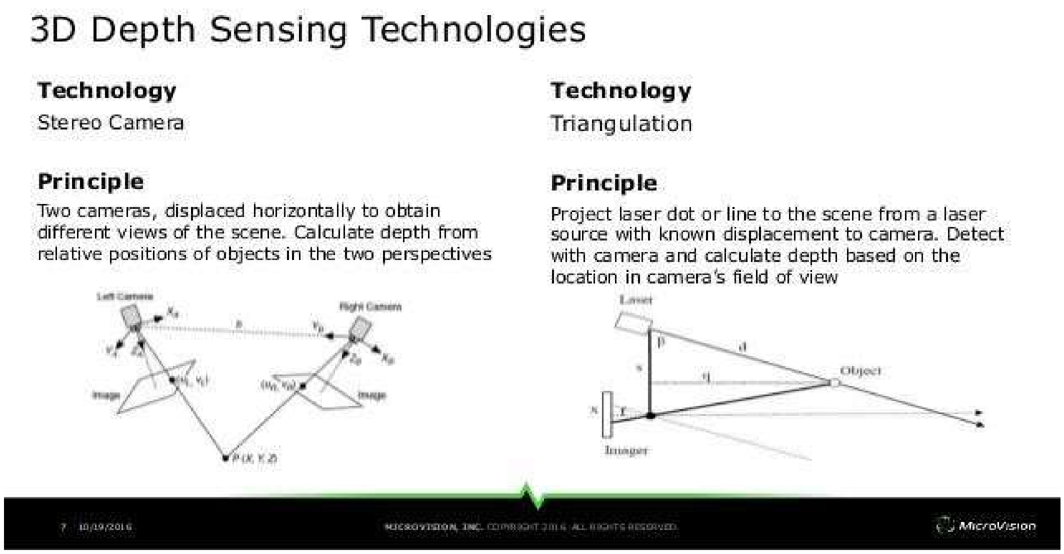

Fig 2.2.b Depth Sensing Technology

2.4 OpenNi:

OpenNI or Open Natural Interaction is an industry-led non-profit organization and open source software project focused on certifying and improving interoperability of natural user interfaces and organic user interfaces for Natural Interaction (NI) devices, applications that use those devices and middleware that facilitates access and use of such devices facial features.

The OpenNI framework provides a set of open source APIs. These APIs are intended to become a standard for applications to access natural interaction devices. The API framework itself is also sometimes referred to by the name OpenNI SDK.

The APIs provide support for

- Voice and voice command recognition
- Hand gestures
- Body Motion Tracking

2.5 Processing Language:

Processing is an open-source graphical library and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching non-programmers the fundamentals of computer programming in a visual context. Processing uses the Java language, with additional simplifications such as additional classes and aliased mathematical functions and operations. It also provides a graphical user interface for simplifying the compilation and execution stage.

2.6 Keras:

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.

CHAPTER 3: ALGORITHMS:

3.1 Depth Calibration:

Depth Calibration is the process of mapping the depth value onto its real world RGB counter path. This problem arises because of the fact that the cameras as physically separated but are pointing to same real-world points.
This method requires finding the intrinsic parameters of depth camera and calculating the real-world point to pixel relationship using the camera matrix.

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{f}{x_3} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

where (x1, x2, x3) are the 3D coordinates of P relative to a camera centered coordinate system, (y1, y2) are the resulting image coordinates, and f is the camera's focal length for which we assume f > 0.

3.2 Convolutional Neural Networks:

A convolutional neural network (CNN, or ConvNet) is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. There are four main steps in CNN: convolution, pooling, activation and fully connectedness

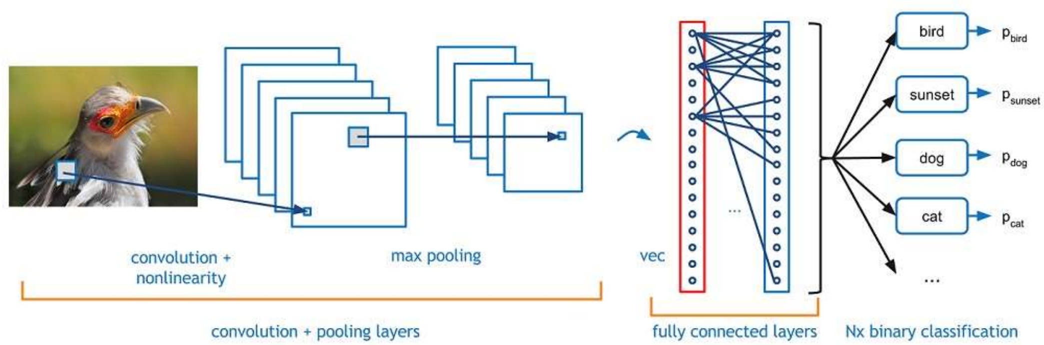

Fig 3.1 Convolutional Neural Network

3.2.a Convolution:

In mathematics (in particular, functional analysis) convolution is a mathematical operation on two functions (f and g) that produces a third function expressing how the shape of one is modified by the other. The term convolution refers to both the result function and to the process of computing it. It is defined as the integral of the product of the two functions after one is reversed and shifted. And the integral is evaluated for all values of shift, producing the convolution function.

$$f(t) * g(t) \triangleq \underbrace{\int_{-\infty}^{\infty} f(\tau)g(t-\tau)\, d\tau}_{(f*g)(t)},$$

3.2.b Pooling:

Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarized features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

The pooling operation is specified, rather than learned. Two common functions used in the pooling operation are:

- Average Pooling: Calculate the average value for each patch on the feature map.
- Maximum Pooling (or Max Pooling): Calculate the maximum value for each patch of the feature map.

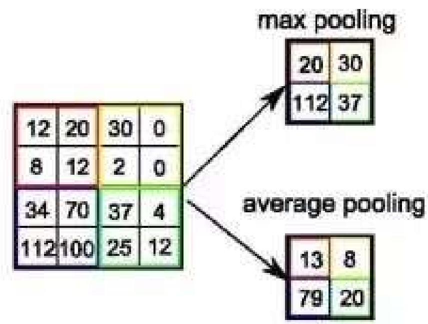

Fig 3.2 Max pooling vs average pooling

3.2.c Activation:

The activation function of a node defines the output of that node given an input or set of inputs. A standard integrated circuit can be seen as a digital network of activation functions that can be "ON" (1) or "OFF" (0), depending on input. This is similar to the behavior of the linear perceptron in neural networks. However, only nonlinear activation functions allow such networks to compute nontrivial problems using only a small number of nodes.

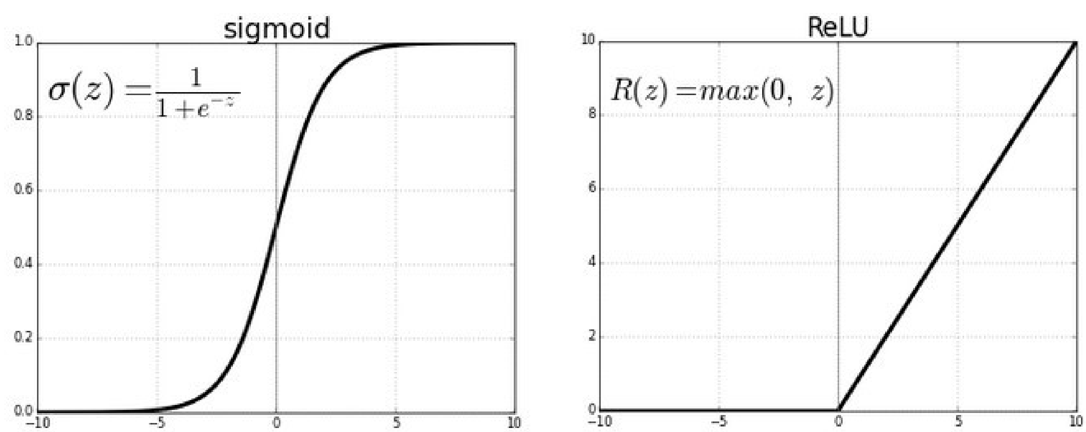

Fig 3.3 Widely used activation functions

3.2.d. Fully Connected:

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

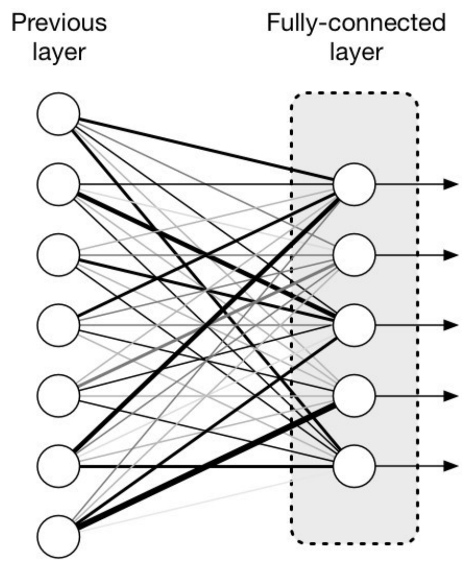

Fig 3.4 Fully Connected Layer

3.3 Transfer Learning:

Transfer learning (TL) is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For example, knowledge gained while learning to recognize cars could apply when trying to recognize trucks. This area of research bears some relation to the long history of psychological literature on transfer of learning, although formal ties between the two fields are limited. From the practical standpoint, reusing or transferring information from previously learned tasks for the learning of new tasks has the potential to significantly improve the sample efficiency of a reinforcement learning agent.

CHAPTER 4: IMPLEMENTATION:

4.1 Outline:

Fig 4.1 Project outline

4.2 Data Acquisition

For our project we have created Sign Language Numbers data set consisting of 10 classes each with 15 samples. The classes correspond to hand sign gestures from 0 to 9 thereby totaling to 10 classes.

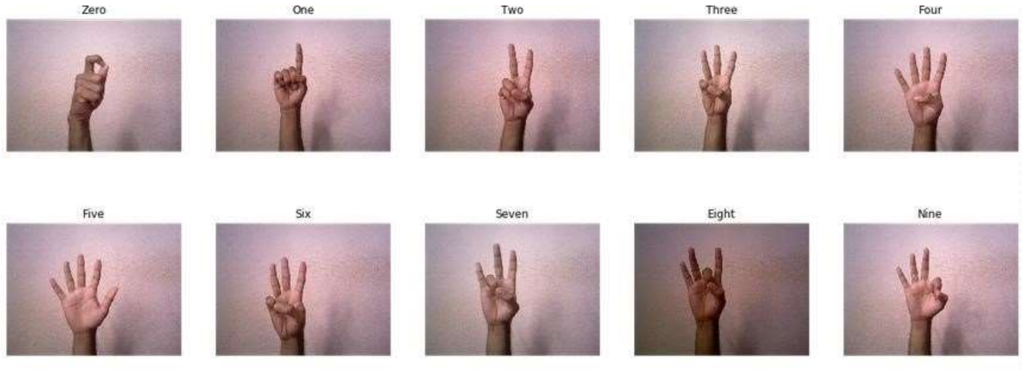

Fig 4.2 RGB Images of Numbers in ISL

The process of data acquisition involved a program based on Java Processing Library with which the data is collected. The so collected data included RGB and Depth images saved in JPG formats with each image dimensions 640 * 480 pixels. In addition to these the 3D positional data x, y, z is stored in text files for all 10 classes in consideration.

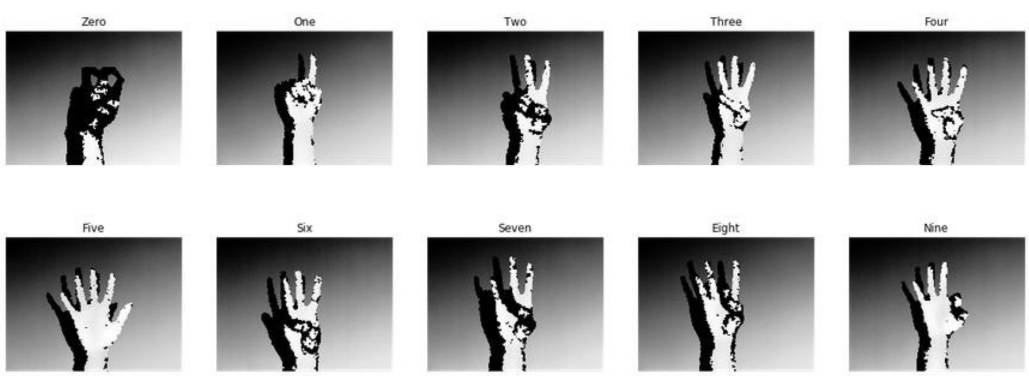

Fig 4.3 Depth Images of Numbers in ISL

4.3 Data Preprocessing:

The data preprocessing involved removing background and labelling the data with their corresponding data class names. Although the background removed images look promising, while training the data there was no significant impact and hence the previous are only used.

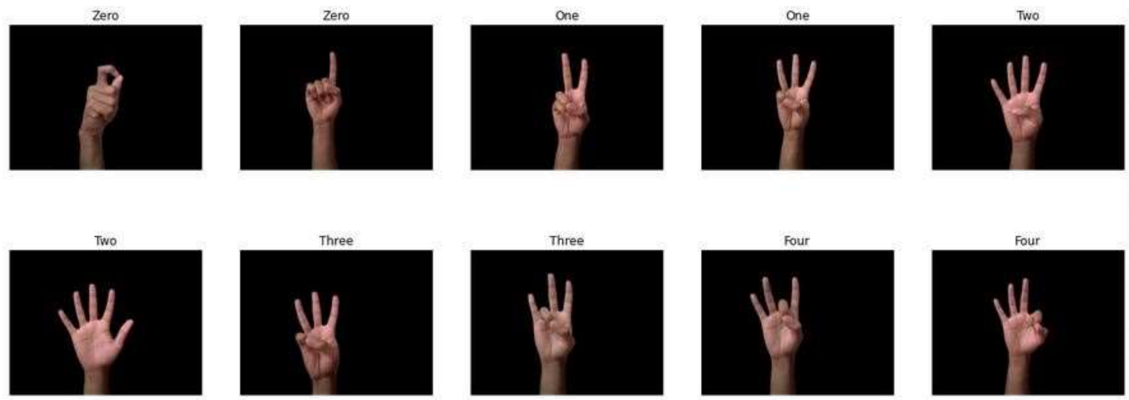

Fig 4.3 Background removed Images of Numbers in ISL

The generation of 4dimensional tensor involved usage of depth calibration techniques so that RGB and corresponding depth can be mapped. This was simply achieved by removing 8 pixels from depth image and rgb image so that all the points are perfectly aligned. The range of depth values stored in the files is from 0 to 255 which are normalized original distances in cm.

4.4 Training:

The training was completely done in a cloud platform that provided GPU acceleration for faster computations. The training was separately done on 3 different types of data – RGB alone, Depth Image alone, Depth + RGB. The networks used comprise VGG and Inception networks at different situations. It is observed that VGG16 network out performed the Inception model in our case for both RGB and Depth images (standalone). For training on Depth + RGB data a custom neural network was build whose architecture is derived from trial and error.

4.4.a Training on RGB and Depth images as separate entities:

The VGG networks are used for training on standalone images in both the cases of train data being RGB and Depth images. Transfer Learning is used in this where pre trained network provided by google is used for training.

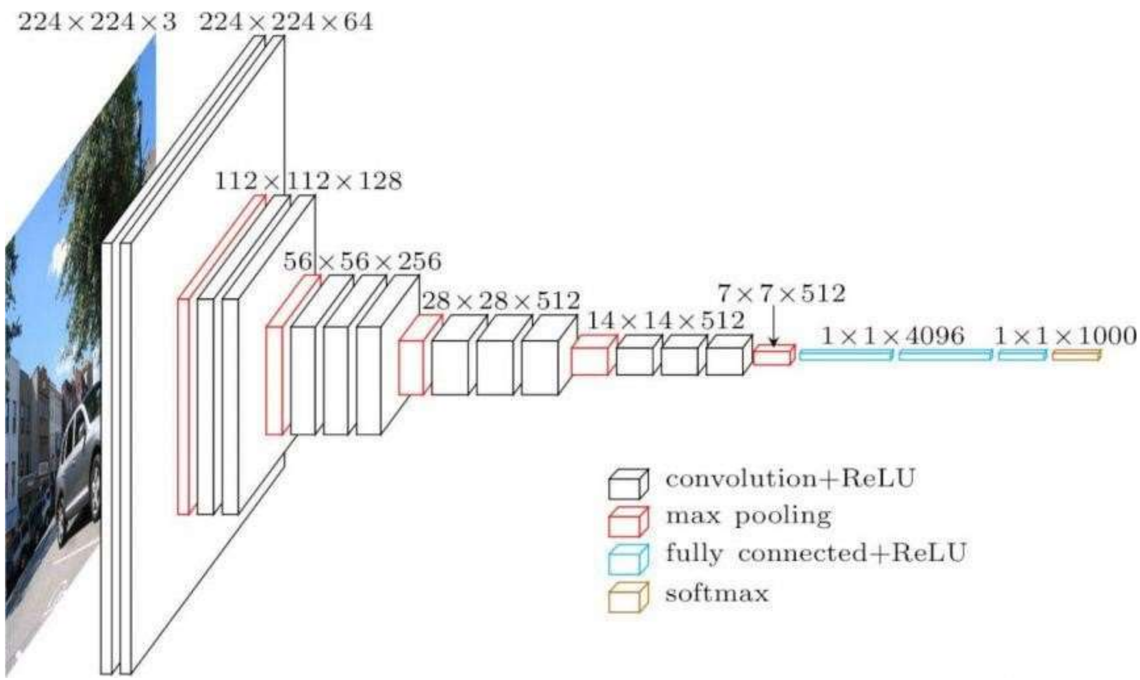

Fig 4.4 VGG16 Network Architecture

The only exception being the input and output layers. The inner layers are nontrainable and hence the only trainable parameters are the input layer and output layer weights. The input format is 480 * 640 * 3 (width, height, RGB) and the output format is a list of 10 values where each value represents the confidence level corresponding to that class index.

Trainable params are 5,130 and the remaining parameters are locked during the entire process. We have obtained a highly accurate model giving rise to an overfit model.

4.4.b Training on RGB and Depth images as combined entity:

In this case we cannot use a pre-trained network because most of the networks available are compatible only with RGB data which means the input layer can only support width * height * (number of channels = 3) format. But what we require is a network capable of handling width * height * (number of channels = 4) data format and for this very reason we have chosen to build our own network.

The input data format is 480(height) * 632(width) * 4(R+G+B+D), observe that the width is reduced by 8 pixels from its original files owing to the fact that depth information had to be mapped onto pixel values. The real-world depth and pixel values of depth image are correlated because the depth image is normalized form of real-world depth in cm.

Layer	Dimensions
Input Layer	Input: (None, 480, 632, 4) Output: (None, 480, 632, 4)
Conv2D	Input: (None, 480, 632, 4) Output: (None, 478, 630, 64)
Maxpooling2D	Input: (None, 478, 630, 64) Output: (None, 239, 315, 64)
Dropout	Input: (None, 239, 315, 64) Output: (None, 239, 315, 64)
Conv2D	Input: (None, 239, 315, 64) Output: (None, 237, 313, 32)

Flatten	Input: (None, 237, 313, 32) Output: (None, 2373792)
Dense	Input: (None, 2373792) Output: (None, 10)

CHAPTER 5: RESULTS AND DISCUSSIONS

We have two different training approaches to the same dataset and hence we end up with different results. There are many metrics used to evaluate the performance of classification problems. We have used confusion Matrix here.

5.1 Results of RGB and Depth images as separate entities:

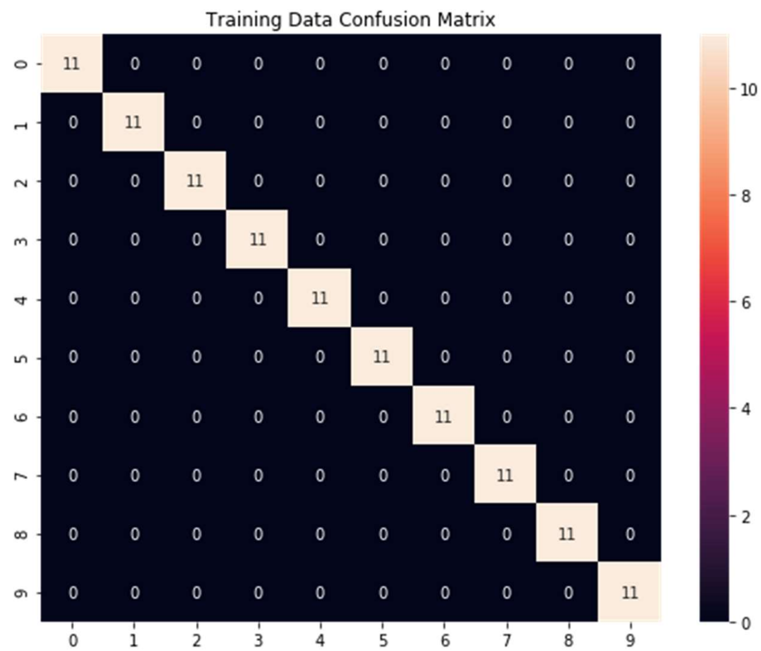

Fig 5.1 Confusion Matrix of RGB train data

The above figure shows a diagonal matrix which proves that the model suffers from overfitting problem with an accuracy of 100%. It may seem obvious to think that it is great success but in-reality the model has lost its ability to generalize. This can be overcome using Data Augmentation techniques and early stopping. A similar result was obtained on Depth images as well meaning to say that the VGG16 network was overfitting the data.

5.2 Results of RGB and Depth images as combined entity:

In order to avoid the previously stated overfitting problem, with this data we have applied early stopping technique to tackle the problem. We also include additional metrics which give us detailed insights of the model performance.

5.2.a Accuracy:

The plot shows that as the number of epochs are increasing the accuracy of both training and validation is also increasing. The crossover point is where the early stopping kicks in and stops the training at that point.

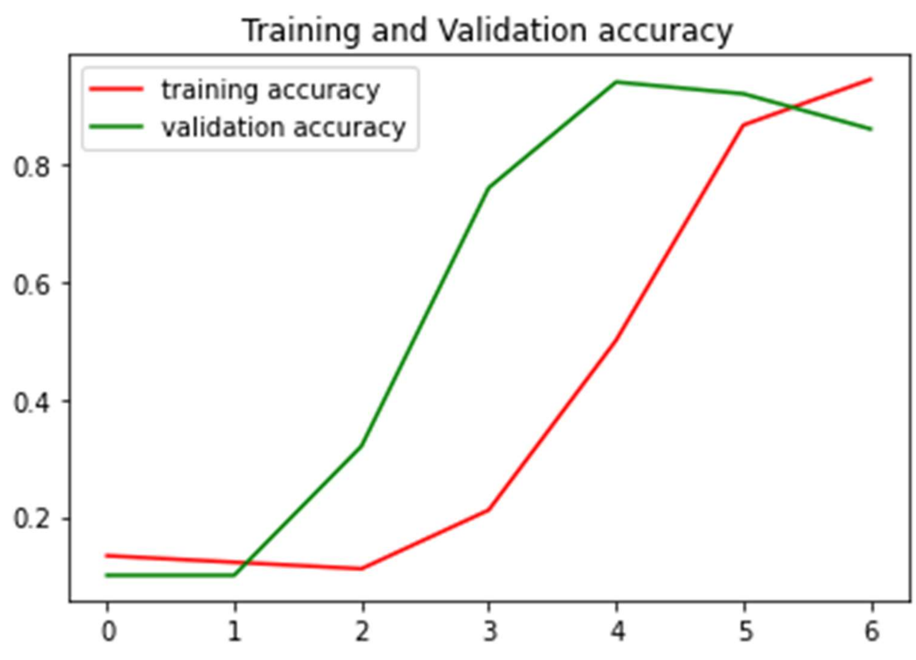

Fig 5.2 Accuracy plot

5.2.b Loss:

The loss used to calculate here is categorical cross entropy which can be seen decreasing as number of epochs are increasing.

Fig 5.3 Loss plot

Initially the loss was very high and as number of epochs increasing, we can see a drastic drop in loss.

5.2.c Confusion Matrix:

From the confusion matrix we can calculate the accuracy to be about 95.5 % and observe that only few samples are mis labeled.

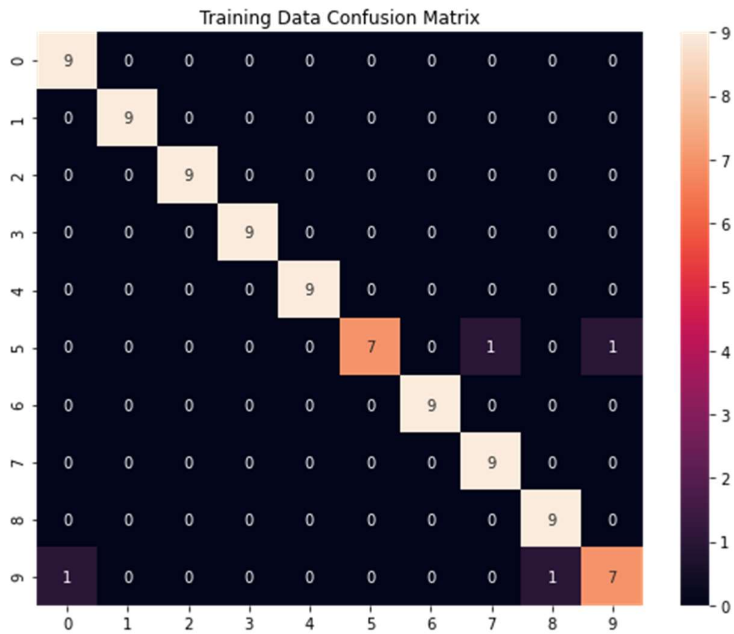

Fig 5.4.a Confusion Matrix

In order to verify its generalization property, we have to test in on completely unseen data samples. We see that the accuracy is 80 %.

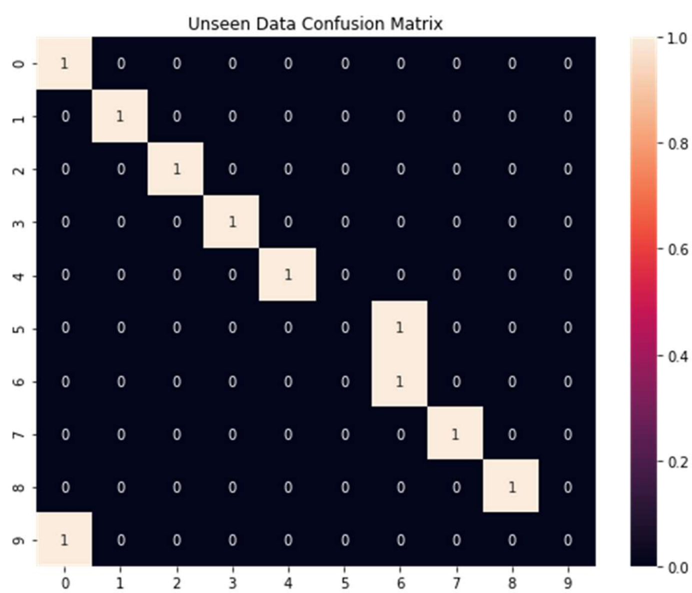

Fig 5.4.b Confusion Matrix

CHAPTER 6: CONCLUSION

6.1 Conclusion:

We aimed to solve the problem of hand sign detection and have presented a way of analyzing RGB and Depth information separately and how they can be used in combination to build a robust Hand sign detection system.

The approach includes both the conventional and non-conventional methods which pave way for next generation human and computer interactions.

6.2 Scope for Future Work:

- Can be extended to a real-time recognition system by synchronizing RGB and depth frames.
- Applications in 3D Modelling systems.
- Application in the field of Virtual Reality and gaming fields.

References

1. M. Ebrahim Al-Ahdal and M. T. Nooritawati, "Review in Sign Language Recognition Systems," *2012 IEEE Symposium on Computers & Informatics (ISCI)*, Penang, 2012, pp. 52-57, doi: 10.1109/ISCI.2012.6222666.

2. A. Kumar, K. Thankachan and M. M. Dominic, "Sign language recognition," *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, Dhanbad, 2016, pp. 422-428, doi: 10.1109/RAIT.2016.7507939.

3. Primesense Specifications https://www.i3du.gr/pdf/primesense.pdf

4. T. Q. Vinh and N. T. Tri, "Hand gesture recognition based on depth image using kinect sensor," *2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, Ho Chi Minh City, 2015, pp. 34-39, doi: 10.1109/NICS.2015.7302218.

5. Arief Ramadhani, Achmad Rizal, Erwin Susanto, W. Martiningsih, R. Wiryadinata, S. Praptodiyono, M.I. Santoso, I. Saraswati, "Development of Hand Gesture Based Electronic Key Using Microsoft Kinect", *MATEC Web of Conferences*, vol. 218, pp. 02014, 2018.

6. Open Natural Interactions https://en.wikipedia.org/wiki/OpenNI

7. Processing https://en.wikipedia.org/wiki/Processing_(programming_language)

8. OpenNI and NITE wrapper https://code.google.com/archive/p/simple-openni/

9. Keras https://en.wikipedia.org/wiki/Keras

10. *West, Jeremy; Ventura, Dan; Warnick, Sean (2007). "Spring Research Presentation: A Theoretical Foundation for Inductive Transfer". Brigham Young University, College of Physical and Mathematical Sciences. Archived from the original on 2007-08-01. Retrieved 2007-08-05.*

11. C. Liang, S. Su and M. Chen, "Non-pre-process calibration of depth image based on fuzzy c-mean," *2017 International Conference on System Science and Engineering (ICSSE)*, Ho Chi Minh City, 2017, pp. 125-128, doi: 10.1109/ICSSE.2017.8030850.

12. J. B. Kulkarni and C. M. Sheelarani, "Generation of Depth Map Based on Depth from Focus: A Survey," *2015 International Conference on Computing Communication Control and Automation*, Pune, 2015, pp. 716-720, doi: 10.1109/ICCUBEA.2015.146.

13. S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

14. Karen Simonyan, & Andrew Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

15. S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, 2015, pp. 730-734, doi: 10.1109/ACPR.2015.7486599.

BIO-DATA

Name: G. PRABHU NITHIN

Roll No: 16011A0431

Date of Birth: 12/07/1998

Nationality: INDIA

Communication address:

H-No 5-22, Road No 1, Adarsh nagar colony, kalimandir, Rajendra nagar, Hyderabad 500086.

Declaration

I hereby declare that the above-mentioned work and my details are true to the best of my knowledge.

Signature

BIO-DATA

Name: MD YASER ALI TARIQ

Roll No: 16011A0421

Date of Birth: 16/08/1999

Nationality: INDIA

Communication address:

8-1-523/33/A, Flat 301,Sky Homes Apartment, Brindavan Colony , Tolichowki, Hyderabad 500008.

Declaration

I hereby declare that the above-mentioned work and my details are true to the best of my knowledge.

Signature

BIO-DATA

Name: S.APURVA

Roll No: 16011A0403

Date of Birth: 25/10/1998

Nationality: INDIA

Communication address:

1-1-141/4/1 ,Road no 1, Srinivasa Nagar colony, NIZAMABAD

Declaration

I hereby declare that the above-mentioned work and my details are true to the best of my knowledge.

Signature

BIO-DATA

Name: D. SUNANDA

Roll No: 16011A0456

Date of Birth: 01/12/1997

Nationality: AMERICAN

Communication address: 5402 Sri Vishnu's Grande Vista,Sy.No.307/2/A/2,Aditya Hills, Pragathi Nagar, Hyderabad, Telangana 500090

Declaration

I hereby declare that the above-mentioned work and my details are true to the best of my knowledge.

Signature