BI-DIRECTIONAL COMMUNICATION SYSTEM FOR DEAF AND DUMB

[1] Prateek Kumar Sharma, [2]Pratik Sharma, [3]Sumit Joon, [4]Ms. Princy Jain
[1,2,3] B.tech Students, [4]Assistant Professor
Department of Information Technology
Dr. Akhilesh Das Gupta Institute of Technology and Management, New Delhi

Abstract: - Communication is the process of exchanging information, views and expressions between two or more persons, in both verbal and nonverbal manner. Hand gestures are the nonverbal method of communication used along with verbal communication. A more organized form of hand gesture communication is known as sign language. In this language each alphabet of the English vocabulary is assigned a sign. The physically disabled person like the deaf and the dumb uses this language to communicate with each other.

Developing sign language application for deaf people can be very important, as 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 idea of this project is to design a system that can understand the sign language accurately so that the less fortunate people may communicate with the outside world without the need of an interpreter. 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. By keeping in mind the fact that in normal cases every human being has the same hand shape with four fingers and one thumb, this project aims at designing a real time system for the recognition primary of ASL alphabets made using hands.

INTRODUCTION

In general, deaf and dumb people use sign language for communication but they find difficulty in communicating with others who do not understand sign language. So, we need a translator to understand what they speak and communicate with us. Also, we need a system which converts the speech of a normal person to text and a corresponding gesture is displayed on display. So, the whole idea is to build a device that enables a two-way communication between a deaf-dumb person and a normal person.

Sign Language is the means of communication in the deaf and dumb community. As a normal person is unaware of the grammar or meaning of various gestures that are the part of a sign language, it is primarily limited to their families or deaf and dumb community. At this age of technology, it is quintessential to make these people feel part of the society by helping them communicate smoothly. Hence, an intelligent computer system is required to be developed and be taught. The idea is to make computer to understand speech and develop a user friendly human computer interfaces (HCI). Making a computer understand speech, and hum an gestures are some important steps towards it. Gestures are nonverbally exchanged information Person can perform innumerable gesture at a time. Science human gestures are perceived through vision; it is subject of great interest for computer vision researches. Coding of these gestures in to machine language demands a complex programming algorithm.

LITERATURE SURVEY

There are many works related to communication system for deaf and dumb that have been identified and literature review is given in this chapter. Aditi Kalsh et.al., (2013) [1], proposed Sign Language Recognition system. As, hand is of major concern here, many challenges make it difficult as well as complex to recognize a particular gesture. The human hand comprises of numerous associated parts and joints, making it a complex object for input. Also, each hand is of a different size. The majority of the sign make utilization of both the hands together. The speed of both the hands is different which a problem is. Shoaib Ibrahim Shaikh et.al. (2016) [2], proposed performance detection of gesture and motion for deaf and dumb communication", Some signs require contact with the body thus, their recognition becomes an issue. It is difficult to maintain a huge data set for gestures as each number and letter has its own sign. Shraddha R.Ghorpade, et.al., (2015) [3], had proposed a system for deaf and dumb people for communication with outside world. Mute people can use gloves to perform a hand gesture and it will be converted into speech so that normal people can understand their expression. The speech of normal people is converted into text and the equivalent gesture for the speech signal will be displayed. Anchal Sood et.al., (2016) [4] proposed a communication system for deaf and dumb.Gesture recognition system is mainly composed of image acquisition, segmentation followed by morphological erosion and feature extraction. The hand gesture is captured through a webcam. The image is segmented using skin detection algorithm. The desired output is shown in the form of text or speech. This framework is inexpensive and easily accessible. This process is very feasible. If there is any gesture which is in contact with body, hand region can be segmented using bounding boxes for all skin regions and then sorting out the one for hand as different body part is of different size. But it is not always 100% accurate. Mayuresh Kein, Shireen meher et.al., (2013) [5] proposed sign language recognition system. This paper proposes that the robust and efficient method of sign language detection. Instead of using Data gloves or sign language detection, we would be doing the detection by image processing. The main advantage of using image processing over Data gloves is that when a new user is using the system it is not required to be re-calibrated. And also by using a threshold value while converting the image from Gray scale to Binary, this system can be used in any background and is not restricted to be used with Black or White Background.

Sawant Pramada, Deshpande Saylee et.al.,(2013) [6], Presented Intelligent sign language recognition using image processing. The system consists of 4 modules. Image is captured through the webcam. Firstly, the captured Colored image is converted into the gray scale image which intern converted into binary form. The captured image is calculated with respect to X and Y coordinates. The calculated coordinates are then stored into the database in the form if template. The templates of newly created coordinates are compared with existing one. If comparison leads to success then the same will be converted into voice and textual form.

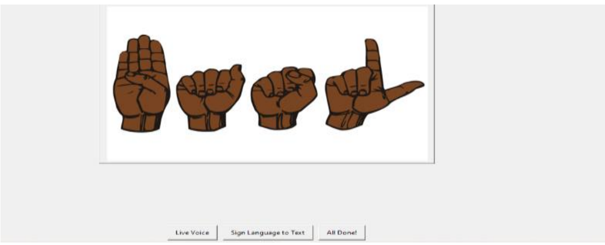
Qing Chen, Nicolas D.Georganas et.al., (2007) [7], proposed real-time vision based hand gesture recognition using haar-like features. In this paper, we proposed a two level approach to recognize hand gestures in real-time with a single webcam as the input device. The low level of the approach deals with the posture recognition with Haar-like features and the AdaBoost learning algorithm. By Haar-like features we can effectively describe the hand posture pattern. By AdaBoost learning algorithm strong classifier can be constructed from combining a sequence of weak classifiers. A parallel cascade structure is implemented to classify different hand postures. From the experiment results, we find this structure can achieve satisfactory real-time performance as well as very high classification accuracy above 90%. The high level hand gestures recognition Deals with the context-free grammar to analyze the syntactic structure based on the detected postures.

PROPOSED SCHEME: **-** A user Interface to select the option of audio to text or sign to text. In sign language to text using webcam, capture the image of the hand to be tested the captured image then undergoes processing that is applying canny edge algorithm to remove all the edge then generate a binary image which is compared with the dataset by CNN and output is displayed. In voice to text Speech Recognition and Py audio are used to capture the audio display the correct output. EasyGUI is used for the interface.

CREATION OF GRAPHICAL USER INTERFACE

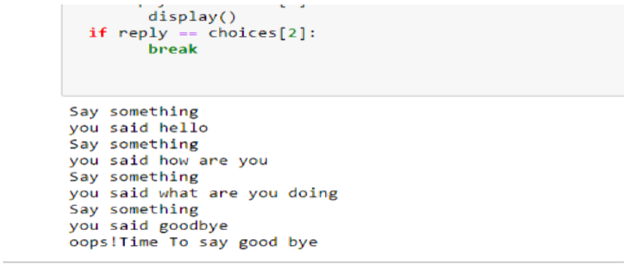
To create a BI-DIRECTION COMMUNICATION SYSTEM FOR DEAF AND DUMB first we need to create a GUI which can effectively interact with the user. For this purpose we will use EasyGUI. EasyGUI is module for simple and easy GUI programming in Python. In this module GUI interactions are invoked by simple function call. We will create a window using EasyGUI which will provide user with and three buttons (LIVE VOICE, SIGN LANGUAGE TO TEXT, ALL DONE).

Creating this window requires user to create a function having while loop and each button calling the respective function associated with it.



SPEECH TO TEXT CONVERSION

A python library known as SPEECH RECOGNITION is required to convert speech to text. Speech recognition, as the name suggests, refers to automatic recognition of human speech. Speech recognition is one of the most important tasks in the domain of human computer interaction. The first component of speech recognition is, of course, speech. Speech must be converted from physical sound to an electrical signal with a microphone, and then to digital data with an analog-to-digital converter. Once digitized, several models can be used to transcribe the audio to text. speech recognition systems rely on what is known as a Hidden Markov Model (HMM). This approach works on the assumption that a speech signal, when viewed on a short enough timescale (say, ten milliseconds), can be reasonably approximated as a stationary process—that is, a process in which statistical properties do not change over time.In a typical HMM, the speech signal is divided into 10-millisecond fragments. The power spectrum of each fragment, which is essentially a plot of the signal's power as a function of frequency, is mapped to a vector of real numbers known as cepstral coefficients. The dimension of this vector is usually small—sometimes as low as 10, although more accurate systems may have dimension 32 or more. The final output of the HMM is a sequence of these vectors. To decode the speech into text, groups of vectors are matched to one or more phonemes—a fundamental unit of speech. This calculation requires training, since the sound of a phoneme varies from speaker to speaker, and even varies from one utterance to another by the same speaker. A special algorithm is then applied to determine the most likely word (or words) that produce the given sequence of phonemes.



```
        display()
    if reply == choices[2]:
        break
```

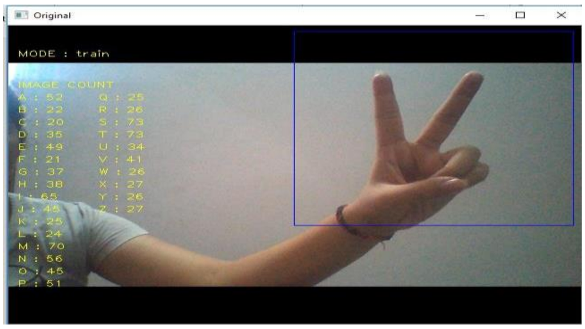
```
Say something
you said hello
Say something
you said how are you
Say something
you said what are you doing
Say something
you said goodbye
oops!Time To say good bye
```

SIGN GESTURE TO TEXT CONVERSION

The first step towards image processing is to acquire the image. The acquired image that is stored in the system windows needs to be connected to the software automatically. This is done by creating an object. With the help of high-speed processors available in computers today, it is possible to trigger the camera and capture the images in real time. The image is stored in the buffer of the object. As has been already discussed, the image is acquired using a simple web camera. Image acquisition devices typically support multiple video formats.

We used Open computer vision (OpenCV) library in order to produce our dataset. Firstly, we captured around 2600 images of each of the symbol in ASL for training purposes and around 650 images per symbol for testing purpose. First, we capture each frame shown by the webcam of our machine. In each frame we define a region of interest (ROI) which is denoted by a blue bounded square. The acquired image is RGB image and needs to be processed before its features are extracted and recognition is made.

We captured images for 2 datasets, i.e., for train and train for all of the 26 alphabets. We also show the number of images each of the alphabets in the mode-test and train has in the camera window too.



```
MODE : train

IMAGE COUNT
A : 52       Q : 25
B : 22       R : 26
C : 20       S : 73
D : 35       T : 73
E : 49       U : 34
F : 21       V : 41
G : 37       W : 26
H : 38       X : 27
I : 65       Y : 26
J : 45       Z : 27
K : 25
L : 24
M : 70
N : 56
O : 45
P : 51
```

Figure 4.1: gesture capturing for train dataset

PREPROCESSING

Main aim of pre-processing is an improvement of the image data that reduce unwanted deviation or enhances image features for further processing. Preprocessing is also referred as an attempt to capture the important pattern which expresses the uniqueness in data without noise or unwanted data which includes cropping, resizing and gray scaling.

Cropping: Cropping refers to the removal of the unwanted parts of an image to improve framing, accentuate subject matter or change aspect ratio.

Resizing: Images are resized to suit the space allocated or available. Resizing image are tips for keeping quality of original image. Changing the physical size affects the physical size but not the resolution.
We capture the gesture that is there within the 'blue box' and as we also resize the captured gesture to (64,64) for both the data sets.

EDGES AND PEAK DETECTION

The edges are detected in the binary image. The Edge Detection block finds the edges in an input image by approximating the gradient magnitude of the image. The block convolves the input matrix with the Sobel, Prewitt, or Roberts's kernel. The block outputs two gradient components of the image, which are the result of this convolution operation. Alternatively, the block can perform a thresholding operation on the gradient magnitudes and output a binary image, which is a matrix of Boolean values. If a pixel value is 1, it is an edge. For Canny, the Edge Detection block finds edges by looking for the local maxima of the gradient of the input image. It calculates the gradient using the derivative of the Gaussian filter. The Canny method uses two thresholds to detect strong and weak edges. It includes the weak edges in the output only if they are connected to strong edges. As a result, the method is more robust to noise, and more likely to detect true weak edges. In this project we have used canny edge detection. The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing.

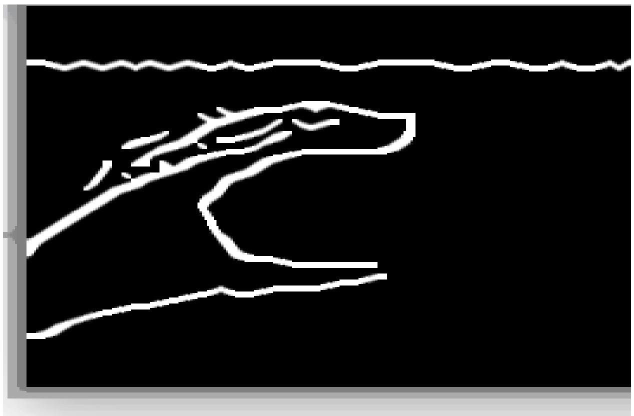


Figure: gesture after canny edge detection

SIGN LANGUAGE RECOGNITION USING CNN

The model used in this classification task is a fairly basic implementation of a Convolutional Neural Network (CNN). As the project requires classification of images, a CNN is the go-to architecture. This model consists of convolutional blocks containing two 2D Convolutional Layers with ReLU activation, followed by Max Pooling and Dropout layers. These convolutional blocks are repeated 3 times and followed by Fully Connected layers that eventually classify into the required categories. The kernel sizes are maintained at 3 X 3 throughout the model.

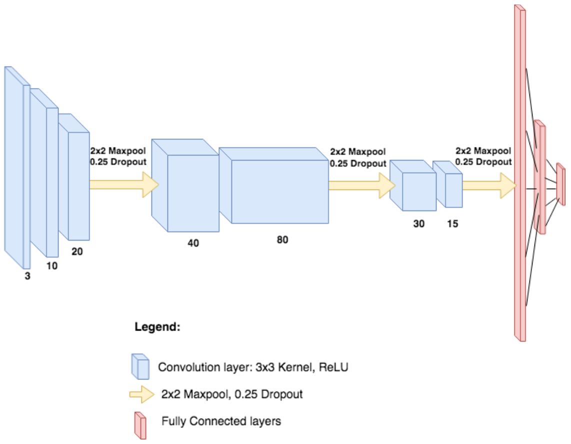


Figure: Model Architecture as implemented using Deep Convolutional Networks

This model was designed to be trained faster and to establish a baseline for problem complexity. This smaller model was built with only one “block” of convolutional layers consisting of two convolutional layers with variable kernel sizes progressing from 5 X 5 to 10 X 10, ReLU activation, and the usual Max Pooling and Dropout. This fed into three fully connected layers which output into the 29 classes of letters. The variation of the kernel sizes was motivated by our dataset including the background, whereas the paper preprocessed their data to remove the background. The design followed the thinking that the first layer with smaller kernel would capture smaller features such as hand outline, finger edges and shadows. The larger kernel hopefully captures combinations of the smaller features like finger crossing, angles, hand location, etc.

IMPLEMENTATION

This data is then preprocessed and then, split into training and validation set using Image in Keras. The complications like over fitting, under fitting, time taken to train the model were solved during implementation by using trial and error method of changing the hyper parameters. The CNN architecture of this project is,

- ➔ Layer 1: inputs = 64, kernel_size=(3, 3), strides=1
- ➔ Dropout: probability = 0.20
- ➔ Layer 2: inputs = 64, kernel_size=(3, 3), strides=1
- ➔ Dropout: probability = 0.20
- ➔ Layer 3: inputs = 64, kernel_size=(3, 3), strides=1
- ➔ Dropout: probability = 0.20
- ➔ Flattening the layer.
- ➔ Dense: inputs = 128
- ➔ Output layer is fully connected layer with softmax activation function.
- ➔ The model is compiled with Adam optimizer and ‘categorical_crossentropy’ as the loss function.

REFINEMENT

Initial result is obtained by building a simple CNN architecture and evaluated. This resulted in more loss (around 0.6) and less accuracy (around 0.8). After the proper tuning of the hyper parameters for a number of times the model’s accuracy increased. For instance, layers inputs were changed from (16, 32, 48), (128, 64, 32) and finally to (64, 64,64), dropout layer was added after every layer for avoiding over fitting, the number of epochs was reduced from 75 to 50 because the accuracy remained the same after the 20thth epoch. After performing all the refinements, the accuracy increased to 99%, the exact value was 0.9886 and the loss was drastically reduced to 0
.0229.

MODEL EVALUATION AND VALIDATION

A fraction 0.20 is used as validation set from the original dataset. After training with the validation set the final accuracy was 0.9886.

Since the test set untouched during the training of the model, it is totally new and unseen by the model and thus the classification is unbiased. Moreover the test set also contained images of hands signs in same backgrounds and were classified well. The status of the training at the last epoch was,

```
Epoch 20/20
2108/2108 [==============================] - 34s
16ms/sample - loss: 0.049
9 - acc: 0.9886 - val_loss: 0.3115 - val_acc: 0.9526
```

This implies that the final training accuracy is 0.9886 and the validation accuracy is 0.9526.

FREE-FORM VISUALIZATION

The accuracy per epoch is plotted in the model accuracy graph which is shown below. It is clearly seen that the accuracy in increases in every epoch. The graph shows both train accuracy (in blue color) and validation accuracy (in orange color).

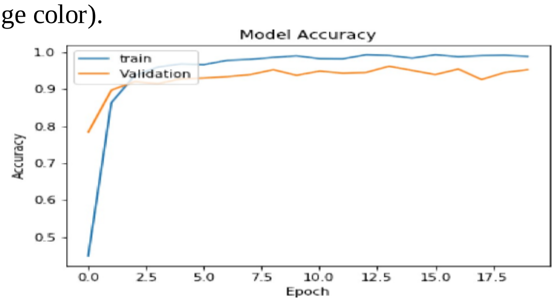


Figure: graph of train and validation accuracy

The loss per epoch is plotted as model loss graph which is shown below. The loss is reduced in every epoch and the loss function used is 'categorical_crossentropy' loss function. The graph shows both train loss (in blue color) and validation loss (in orange color).

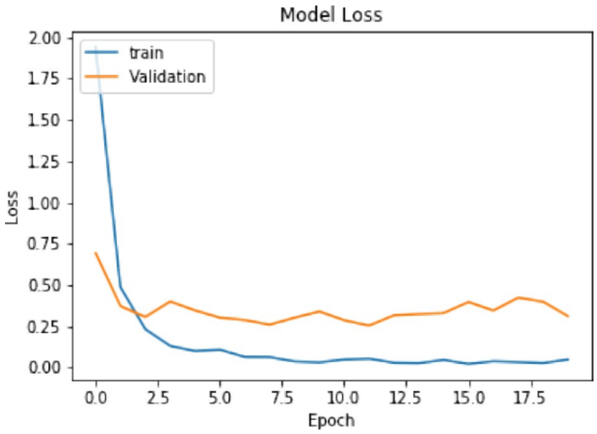
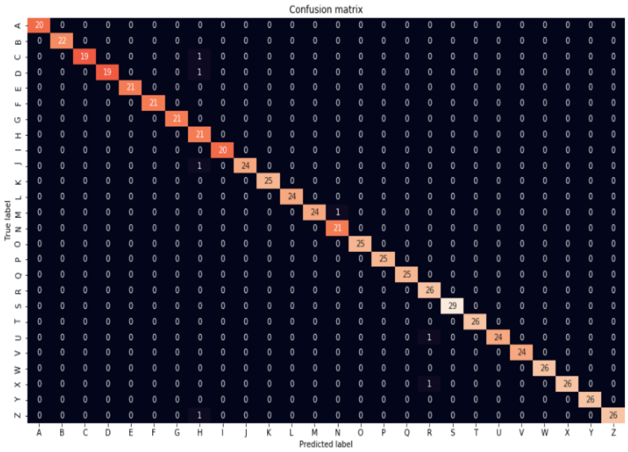


Figure: train and validation loss

EVALUATION METRICS

CONFUSION MATRIX

A **confusion matrix** (also known as an error matrix) is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes.

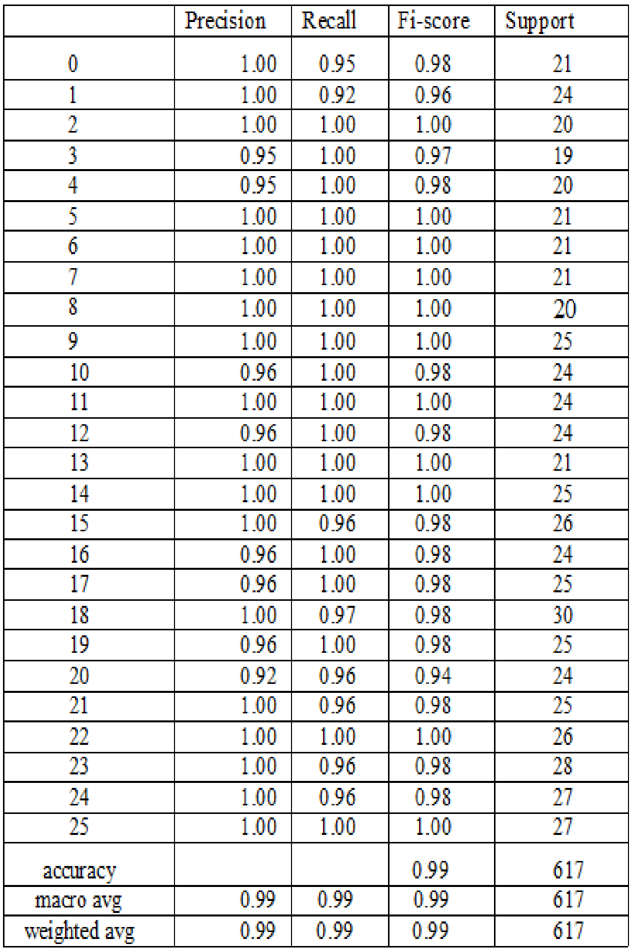


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	19	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	19	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	1	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	24	1	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	24	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	26	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0
Z	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26

Figure: - Confusion matrix

Classification Report

A **Classification report** is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report.



	Precision	Recall	Fi-score	Support
0	1.00	0.95	0.98	21
1	1.00	0.92	0.96	24
2	1.00	1.00	1.00	20
3	0.95	1.00	0.97	19
4	0.95	1.00	0.98	20
5	1.00	1.00	1.00	21
6	1.00	1.00	1.00	21
7	1.00	1.00	1.00	21
8	1.00	1.00	1.00	20
9	1.00	1.00	1.00	25
10	0.96	1.00	0.98	24
11	1.00	1.00	1.00	24
12	0.96	1.00	0.98	24
13	1.00	1.00	1.00	21
14	1.00	1.00	1.00	25
15	1.00	0.96	0.98	26
16	0.96	1.00	0.98	24
17	0.96	1.00	0.98	25
18	1.00	0.97	0.98	30
19	0.96	1.00	0.98	25
20	0.92	0.96	0.94	24
21	1.00	0.96	0.98	25
22	1.00	1.00	1.00	26
23	1.00	0.96	0.98	28
24	1.00	0.96	0.98	27
25	1.00	1.00	1.00	27
accuracy			0.99	617
macro avg	0.99	0.99	0.99	617
weighted avg	0.99	0.99	0.99	617

True Positive (TP): Observation is positive, and is predicted to be positive.

False Negative (FN): Observation is positive, but is predicted negative.

True Negative (TN): Observation is negative, and is predicted to be negative.

False Positive (FP): Observation is negative, but is predicted positive.

Classification Rate/Accuracy:

Classification Rate or Accuracy is given by the relation:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Our accuracy as seen from the classification report is 99%.

Recall:

Recall is the ability of a classifier to find all positive instances. Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (small number of FN).Recall is given by the relation:

$$\textit{Recall} = \frac{TP}{TP + FN}$$

Our recall as seen from the classification report is 99%.

Precision:

Precision is the ability of a classifier not to label an instance positive that is actually negative. To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labeled as positive is indeed positive (small number of FP). Precision is given by the relation:

$$\textit{Precision} = \frac{TP}{TP + FP}$$

Our precision as seen from the classification report is 99%.

F-measure:

The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. The F-Measure will always be nearer to the smaller value of Precision or Recall.

$$\textit{F - measure} = \frac{2*Recall*Precision}{Recall + Precision}$$

Our F1 score as seen from the classification report is 99%.

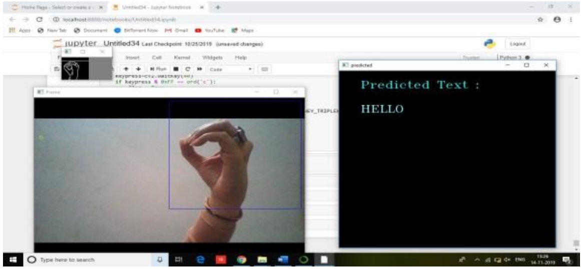
GESTURE RECOGNITION MODEL OUTPUT

The output of the model consists of three windows.

1 - First window takes the gesture created by our hand and predict that alphabet and show that alphabet on the top left corner in yellow color.

2 - Second window shows the gray scale image with canny's edge detection.

3 - The third window shows the string of the output alphabets in the sky blue color.



```
Predicted Text :

HELLO
```

jupyter Untitled34 Last Checkpoint: 10/25/2019 (unsaved changes)
predicted
Type here to search

Figure: Final output

CONCLUSION

Hand gestures are a powerful way for human communication, with lots of potential applications in the area of human computer interaction. Vision based hand gesture recognition techniques have many proven advantages compared with traditional devices. However, hand gesture recognition is a difficult problem and the current work is only a small contribution towards achieving the results needed in the field of sign language gesture recognition. This report presented a vision-based system able to interpret isolated hand gestures from the American Sign Language (ASL).

With the intention of offering a thorough study of neural networks that learn from ASL images and videos, it has been found that some theoretical notions that work well in related tasks are not as adequately suited for our purpose. Delving into the types of input information constituted an interesting exploration: whereas key point coordinates for the joints and fingers proved successful in the learning process, optical flow did not provide an insight into spatial-temporal manual signs. 3D convolutional layers, on the other hand, were positively validated in the Modelling of short temporal correspondences. In terms of managing the depth or time dimension through the CNN, it was found that gradual fusion meant that the network learnt about the spatial features first and progressively incorporated the temporal features. CNN model has obtained an accuracy of 99 %.

Pre-processing was a consequential aid, particularly ROI extraction, although previous works have noted that ignoring the features outside of the extracted region can hinder the understanding of the context of the information within the ISL. This has been shown to be true, but it is not a substantial concern for the given framework. In any case, the relative position of the hands in the frame would definitely provide relevant contextual information for a larger dictionary.

Many applications can be derived from these results, with some refinements: a real time fingerspelling captioning application, a tool to automatically subtitle ISL videos given the signs' time codes, or a virtual ISL teaching assistant that evaluates the student by comparing to the learnt signs.

FUTURE WORK

Future courses of action can be suggested as follow-up lines of work to this dissertation.

An interesting area is continuous sign recognition, which aims to detect and isolate signs from sequences. This involves recognizing movement epenthesis, a term that refers to the transitional motion between signs. The HMM can be trained to classify each sequence as a sign or epenthesis, according to a probability distribution computed for a two-hand gesture. It would be interesting to apply a different approach to ISL data, such as identifying lip movement patterns: Pfizer et al. focus on the openness of the mouth to distinguish between sign and silence, because it is common to mouth the word that is being signed, so there is co-occurrence.

Furthermore, this research could be extended to encompass more factors of expression besides hands. A multi-modal system that incorporates facial and body language would be able to better convey the full meaning and connotations of what is being transmitted through signs. Also, a hybrid RNN and CNN approach may be superior.

Throughout the development of this project, the lack of a great amount of precisely annotated data has posed some issues. Clearly, gathering more samples and strongly supervised data would assist the progress of developing automatic ASL recognition systems.

This method for individual gestures can also be extended for sentence level sign language. Also, for future work one can focus on combining the CNN and RNN models into a single model because a hybrid RNN and CNN approach may be superior.

REFERENCES

1. Aditi Kalsh and N.S. Garewa, “Sign Language Recognition system,”,International Journal of computational Engineering Research, vol, 03 Issue, 6,June,2013
2. Shoaib Ibrahim Shaikh, Ismail Mehmood Memom, SanmayJayaram Shetty, Aziz SaifuddinVakanerwala, “performance Detection of Gesture and Motion for Deaf and Dumb communication”, International Journal of Innovative Research in science, Engineering and Technology(IJIESET), volume 5, Issue 2, 15th February 2016.
3. Shraddha R.Ghorpade, Surendra k.waghmare ,“Full Duplex Communication System for Deaf and Dumb people” , International Journal of Emerging Technology and Advanced Engineering, volume 5, Issue 5, May 2015.
4. Anchal Sood and Anju Mishra , “A Communication System for Deaf and Dumb”International Conference on Reliability, InfocomTechnologies and optimization (ICRITO),sep. 7-9, 2016
5. Mayuresh Kein, Shireen Meher and Aniket Marathe,”sign language recognition system”International journalof scientific & Engineering Research, volume4, Issue 12, Decmber- 2013.
6. Sawant Pramada, Deshpande Savlee and Nale Pranita““Intelligent sign language recognition using image processing““IOSR Journal of Engineering (IOSRJEN), volume3,Issue2 (Feb.2013).
7. Qing Chen, Nicolas D.Georganas and Emil M.Petriu, Real-time Vision based Hand Gesture Recognition Using Haar-like Features” IEEE Instrumentation and Measurement Technology Conference, Jun-2017.
8. Ms.Rashmi D.Kyatanavar Prof.P.R.Futane. Comparative study od sign Language Recognition Systems, International Journal of Scientific and Reasearch Publications, volume2, Issue6 June2012 ISSN 2250-3153.
9. Ravikiran J, Kavi Mahesh, SuhasMahishi, Dheeraj R, Sudheender S, Nitin V Pujari, Finger Detection for Sign Language Recognition, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, MARCH 18-20, 2009, HongKong.
10. Pallavi Verma, Shimi S. L., Richa Priyadarshani “Design of Communication Interpreter for Deaf and Dumb Person”, International Journal of Science and Research, Volume 4 Issue 1, January 2015
11. Richard Watson, “Gesture recognition techniques”, Technical report, Trinity College, Department of Computer Science, Dublin, July, Technical Report No. TCD-CS-93-11, 1993
12. Ms. Rashmi D. Kyatanavar, Prof. P. R. Futane, “Comparative Study of Sign Language Recognition Systems”, Department of Computer Engineering, Sinhgad College of Engineering, Pune, India International Journal of Scientific and Research Publications, Volume 2, Issue 6, June 2012 ISSN 2250- 3153

Books:

1. “Digital Image Processing” (2nd Edition) Rafael C. Gonzalez (Author), Richard E. Woods (Author) Publication Date: January 15, 2002 | ISBN-10: 0201180758 | ISBN-13: 978-0201180756.