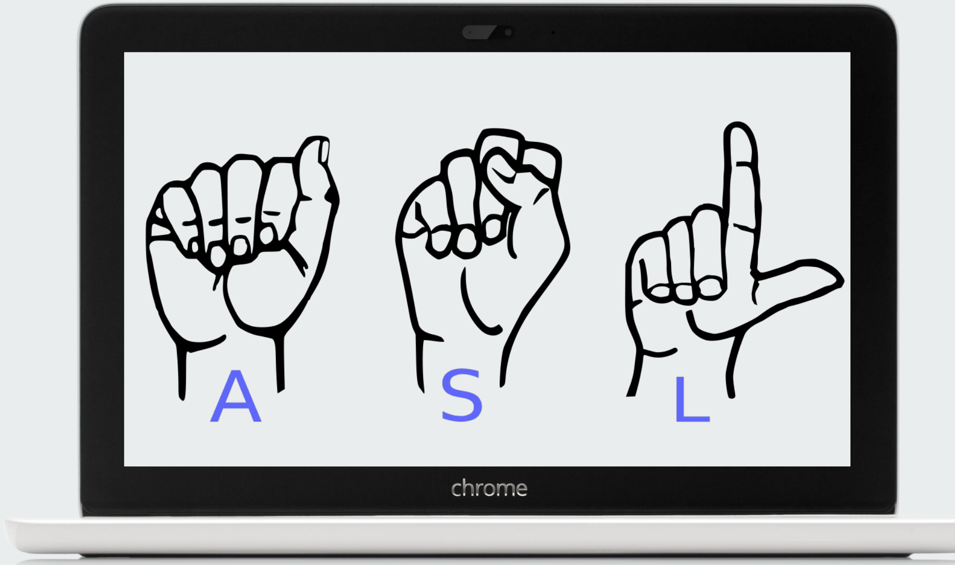
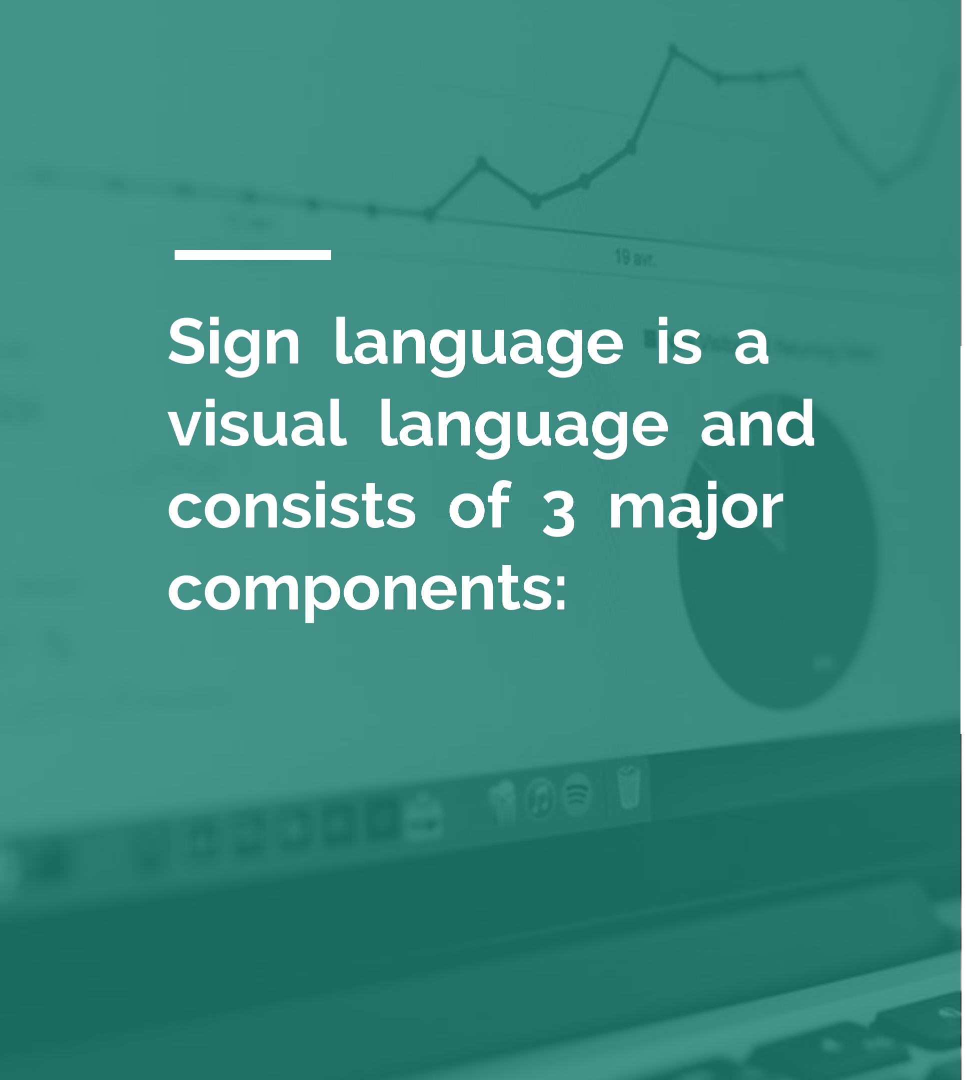
Conversion of Sign Language to Text

For Dumb and Deaf

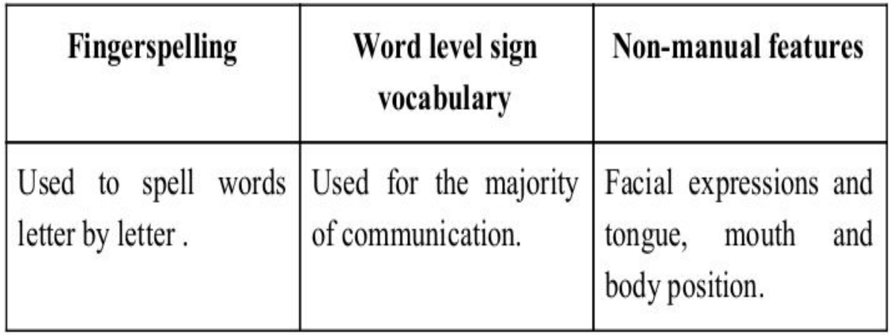


Abstract

Our project aims to create a computer application and train a model which when shown a real time video of hand gestures of American Sign Language shows the output for that particular sign in text format on the screen.

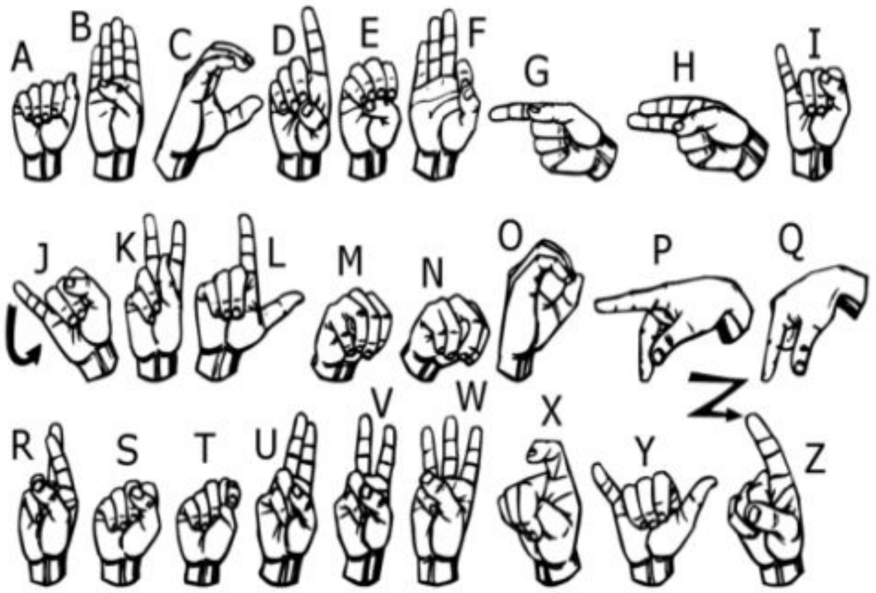


Sign language is a visual language and consists of 3 major components:



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.

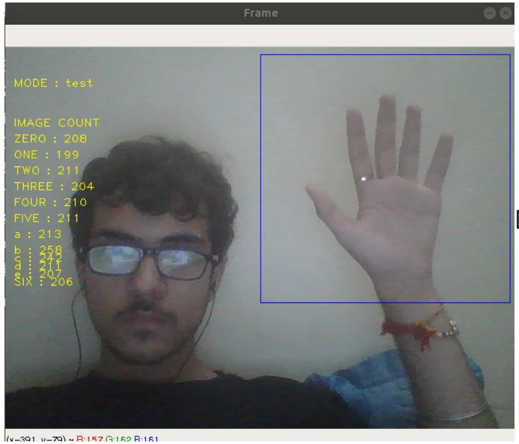
We implemented 27 symbols(A-Z, blank) of ASL in our project.



Methodology

How we generated data set and did Data Preprocessing ?

Capturing Raw Image



```
MODE : test

IMAGE COUNT
ZERO : 208
ONE : 199
TWO : 211
THREE : 204
FOUR : 210
FIVE : 211
a : 213
b : 258
c : 242
d : 211
e : 207
SIX : 206
```

```
(x=391, y=79) ~ R:157 G:162 B:161
```

Gray Scale Image

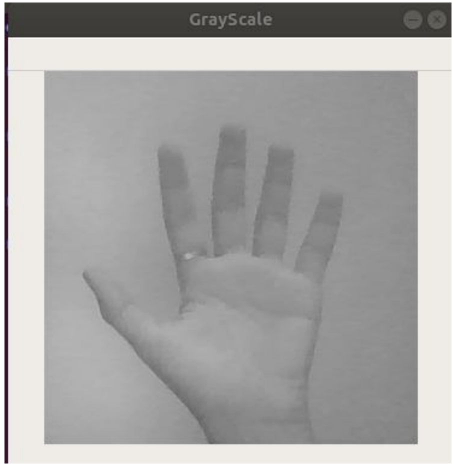
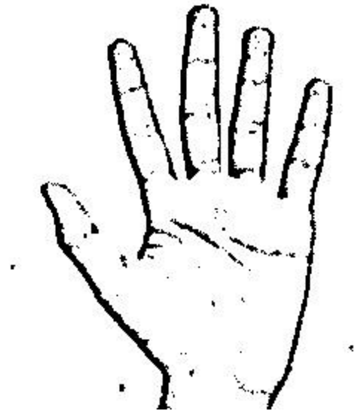


Image Post Gaussian Blur



Why we Created our own Dataset ?

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

Gesture Classification

$$\begin{tikzcd} \arrow[r] & \end{tikzcd}$$

Layer 1

Classify between 27 Symbols

$$\begin{tikzcd} \arrow[r] & \end{tikzcd}$$

Layer 2

Classify between Similar Symbols

Algorithm Layer 1:

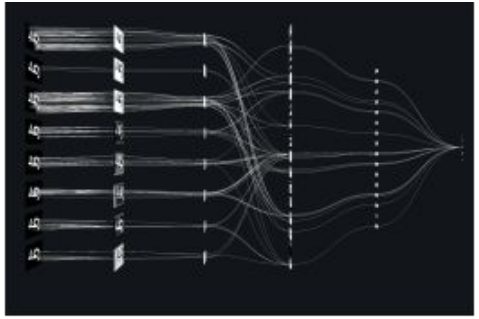
1. Apply gaussian blur filter and threshold to the frame taken with opencv to get the processed image after feature extraction.
2. This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.
3. Space between the words are considered using the blank symbol.

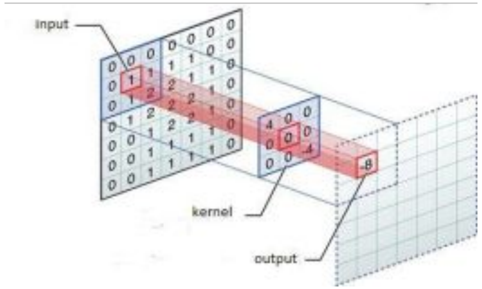
Algorithm Layer 2:

- We detect various sets of symbols which show similar results on getting detected.
- We then classify between those sets using classifiers made for those sets only.
- In our testing we found that following symbols were not showing properly and were giving other symbols also :
    1. For D : R and U
    2. For U : D and R
    3. For I : T, D, K and I
    4. For S : M and N

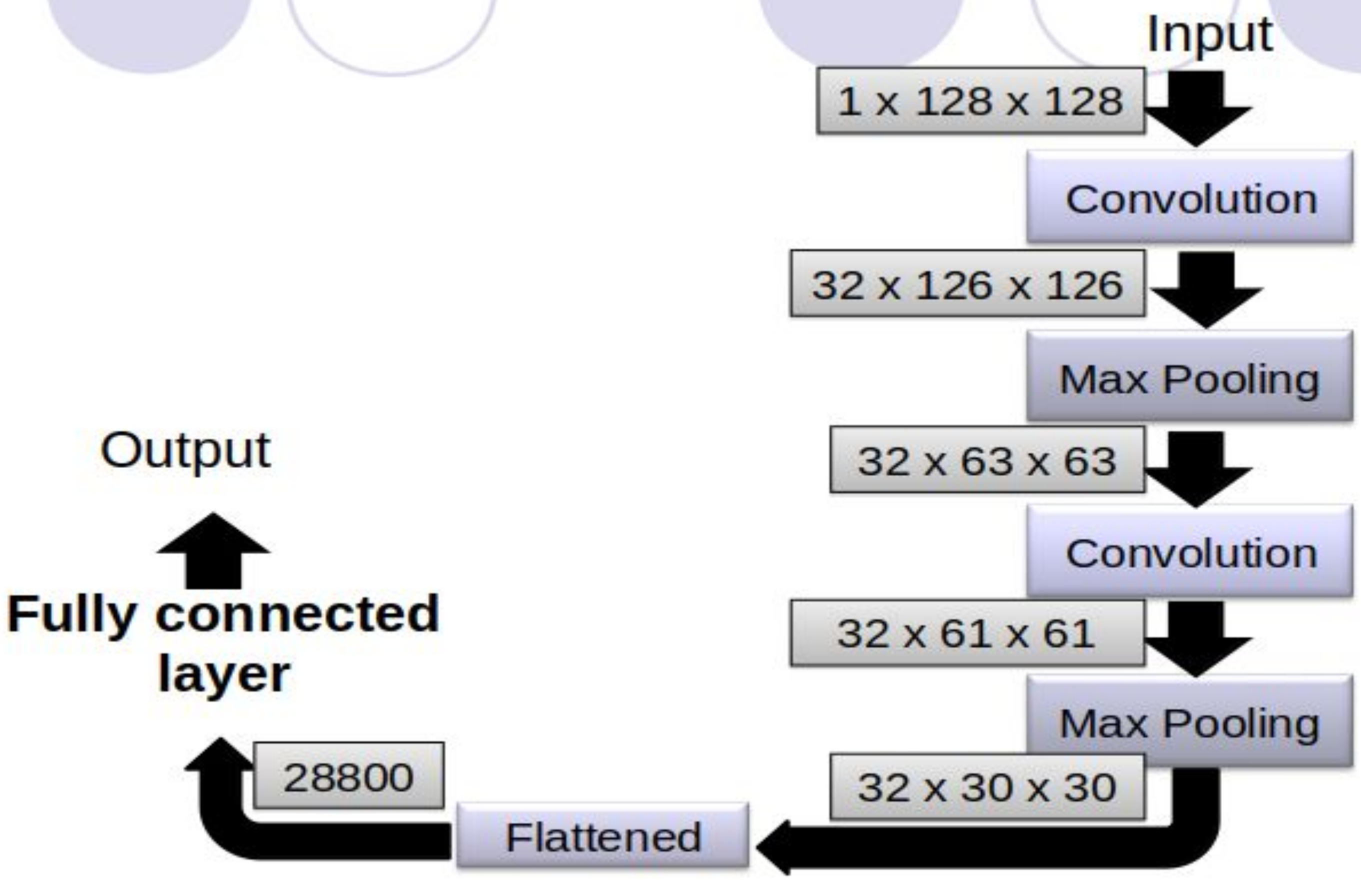
Convolutional Neural Networks

- CNNs consist of multiple convolutional layers each layer containing numerous “filters” which perform feature extraction.
- Initially these “filters” are random and by training, the feature extraction gets better by better.
- It’s primarily used for image classification.





Our CNN Classifier Model


Input
1 x 128 x 128
Convolution
32 x 126 x 126
Max Pooling
32 x 63 x 63
Convolution
32 x 61 x 61
Max Pooling
32 x 30 x 30
Flattened
28800
Fully connected layer
Output

Finger Spelling Sentence Formation

Implementation

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

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

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

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

Autocorrect feature

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

Challenges Faced

- ➢ We couldn’t find a dataset with raw images of all the asl characters so we made our own dataset.
- ➢ Second issue was to select a filter for feature extraction. We tried various filter including binary threshold, canny edge detection, gaussian blur etc. ,of which gaussian blur filter was giving better results.
- ➢ Issues were faced relating to the accuracy of the model we trained in earlier phases which we eventually improved by increasing the input image size and also by improving the dataset.

Software Requirements

- Python 3.6.6
- Tensorflow 1.11.0
- OpenCV 3.4.3.18
- NumPy 1.15.3
- Matplotlib 3.0.0
- Hunspell 2.0.2
- Keras 2.2.1
- PIL 5.3.0

Limitations of our model

- The model works well only in good lighting conditions.

- Plain background is needed for the model to detect with accuracy.

Conclusion

- In this report, a functional real time vision based american sign language recognition for D&M people have been developed for asl alphabets.
- We achieved an accuracy of **95.7%** on our dataset.
- Prediction has been improved after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other.

Future Scope

- ❖ We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms.
- ❖ We are also thinking of improving the preprocessing to predict gestures in low light conditions with a higher accuracy.

Thank You !