ACROPOLIS

Enlightening Wisdom

Acropolis Institute of Technology & Research, Indore

Sign Language Interpreter

Submitted to:

Department of Computer Science and Engineering

ACROPOLIS
Enlightening Wisdom

Supervised by:
Prof. Krupi Saraf

Team Members

1. Mahak Chittoda
2. Mayuri Balchandani
3. Neelesh Vaishnav
4. Pranav Bharadwaj

Project Presentation Outline

- Abstract
- Introduction
- Problem Statement
- Survey of Existing Systems
- Project Objectives
- Requirement Analysis
- Solution Proposed
- Models and Diagrams
- The Implementation
- Testing
- The Outcome Discussion
- Conclusions and Limitations

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.

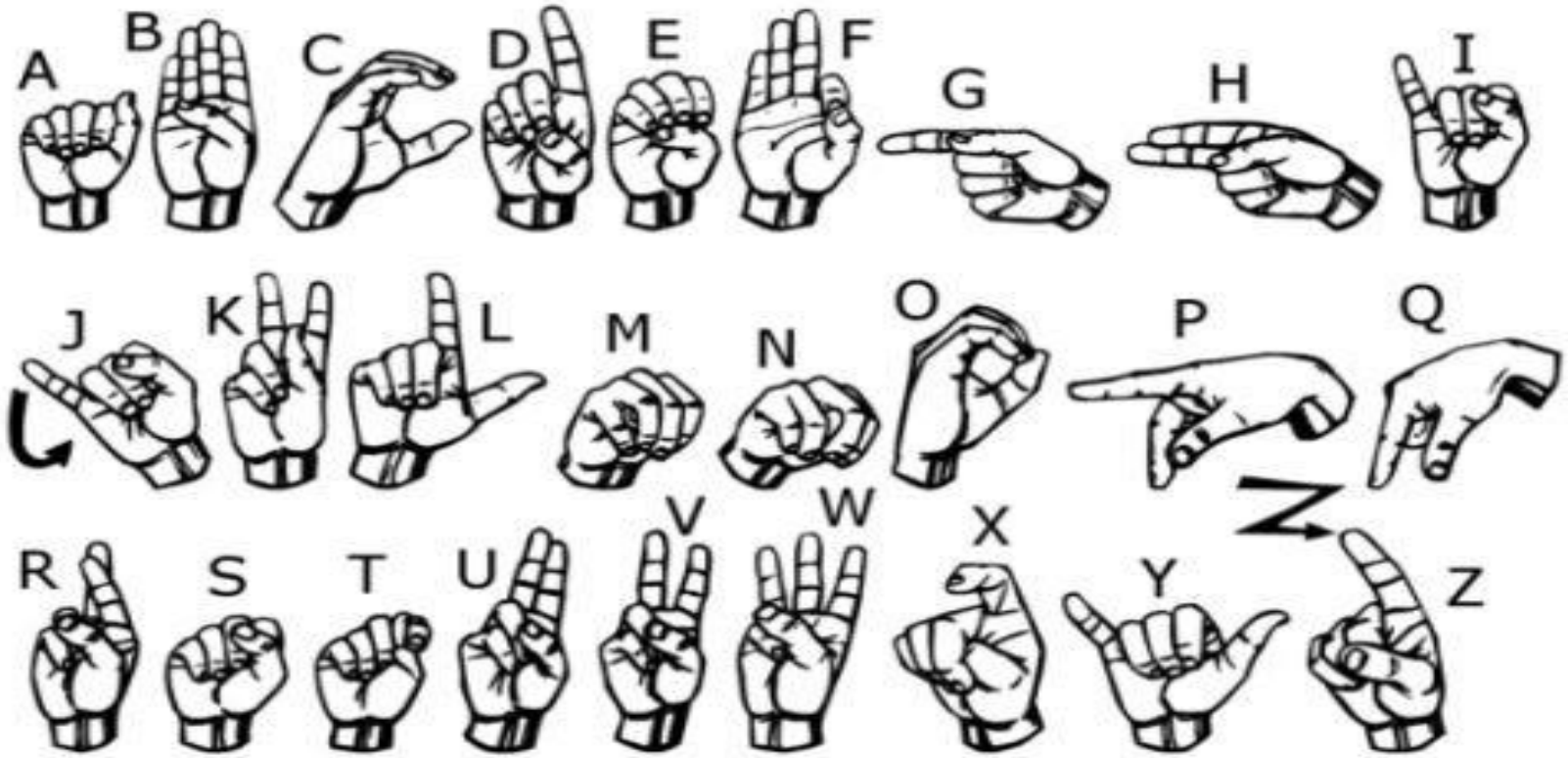
- In our method, the hand is first passed through a filter and after the filter is applied the hand is passed through a classifier which predicts the class of the hand gestures. Our method provides 95.7 % accuracy for the 26 letters of the alphabet.

Introduction

- In our project we primarily focus on producing a model which can recognize 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.

- American sign language is a predominant sign language Since the only disability Deaf and Dumb (hereby referred to as D&M) people have is communication related and since they cannot use spoken languages, the only way for them to communicate is through sign language.

Introduction



Motivation

- For interaction between normal people and D&M people a language barrier is created as sign language structure since it is different from normal text. So, they depend on vision-based communication for interaction.

- If there is a common interface that converts the sign language to text, then the gestures can be easily understood by non-D&M people. So, research has been made for a vision-based interface system where D&M people can enjoy communication without really knowing each other's language.

Motivation

We have also been motivated observing the benefits of physically handicapped people like deaf and dumb. But if any normal human being or an automated system can understand their needs by observing their facial expression then it becomes a lot easier for them to make the fellow human or automated system understand their needs.

The Problem Statement

- Dumb people use hand signs to communicate, hence normal people face problem in recognizing their language by signs made. Hence there is a need of the systems which recognizes the different signs and conveys the information to the normal people.

Survey of Existing Systems

- In existing system the module was developed for dumb person using flex sensor, there user hand is attached with the flex sensors.
- On this module the flex sensor reacts on bend of each finger individually. By taking that value controller starts to react with speech, each flex sensor holds unique voice stored in APR Kit and for each sign it will play unique voice.
- And in other existing system, the work is done only for some alphabets and not for the words or sentences, and accuracy obtained is very low.

Survey of Existing Systems

- **Limitations of Existing Systems:**
- In existing system it's restricted to only 10 voice announcements it may reduce product capacity.
- One of the major problem of the existing system is Dumb person should always carry the hardware with him .
- User can't do any other work with flex sensor on fingers and also sensors should be placed straight .
- The controller may think that the user is giving command and finally it may result in unwanted results and less hardware lifetime.

Objectives

- The main objective is to translate sign language to text/speech. The framework provides a helping-hand for speech-impaired to communicate with the rest of the world using sign language.
- This leads to the elimination of the middle person who generally acts as a medium of translation.
- This would contain a user-friendly environment for the user by providing speech/text output for a sign gesture input.

Requirement Analysis

Functional Requirements:

Functional requirements are features that the system will need in order to deliver or operate.

- Capture face images via webcam or external USB camera.
- Hand gestures on an image must be detected.
- The hand gestures must be detected in bounding boxes.
- System should have a graphic card to process the images and videos on real time.

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

Requirement Analysis

Non-Functional Requirements

Non-functional requirements are set of requirements with specific criteria to judge the systems operation.

- Reliable
- Ease of use
- Secure

Solution Proposed

- American sign language is a predominant sign language Since the only disability D&M people have is communication related and they cannot use spoken languages hence the only way for them to communicate is through sign language.

- Communication is the process of exchange of thoughts and messages in various ways such as speech, signals, behavior and visuals.Deaf and Mute(Dumb)(D&M) people make use of their hands to express different gestures to express their ideas with other people.

- Gestures are the nonverbally exchanged messages and these gestures are understood with vision. This nonverbal communication of deaf and dumb people is called sign language.

Solution Proposed

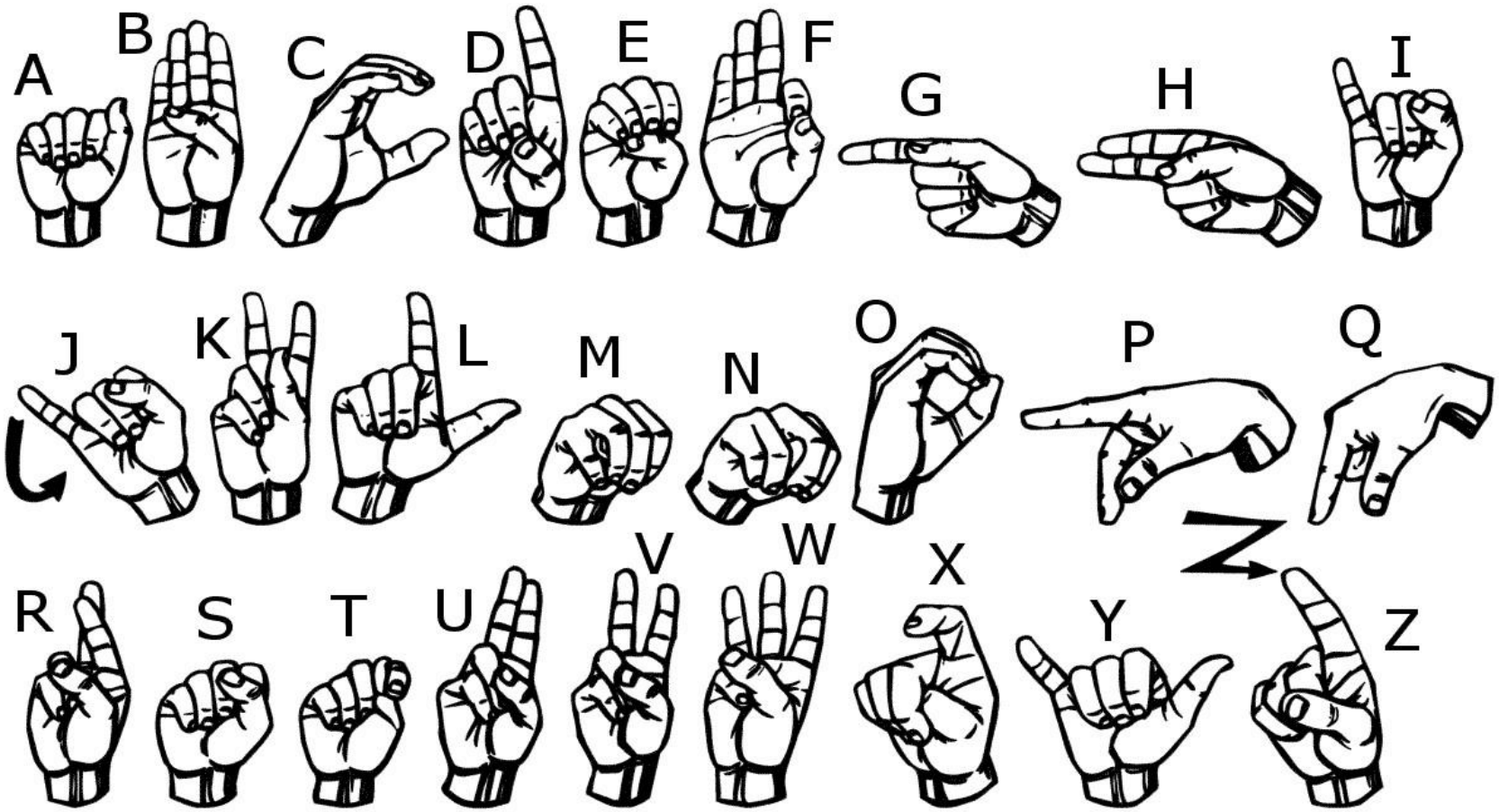
- Sign language is a visual language and consists of 3 major components:

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.

- In this project we basically focus on producing a model which can recognize Fingerspelling based hand gestures in order to form a complete word by combining each gesture.

Solution Proposed

- The gestures we trained are as follows:



Solution Proposed

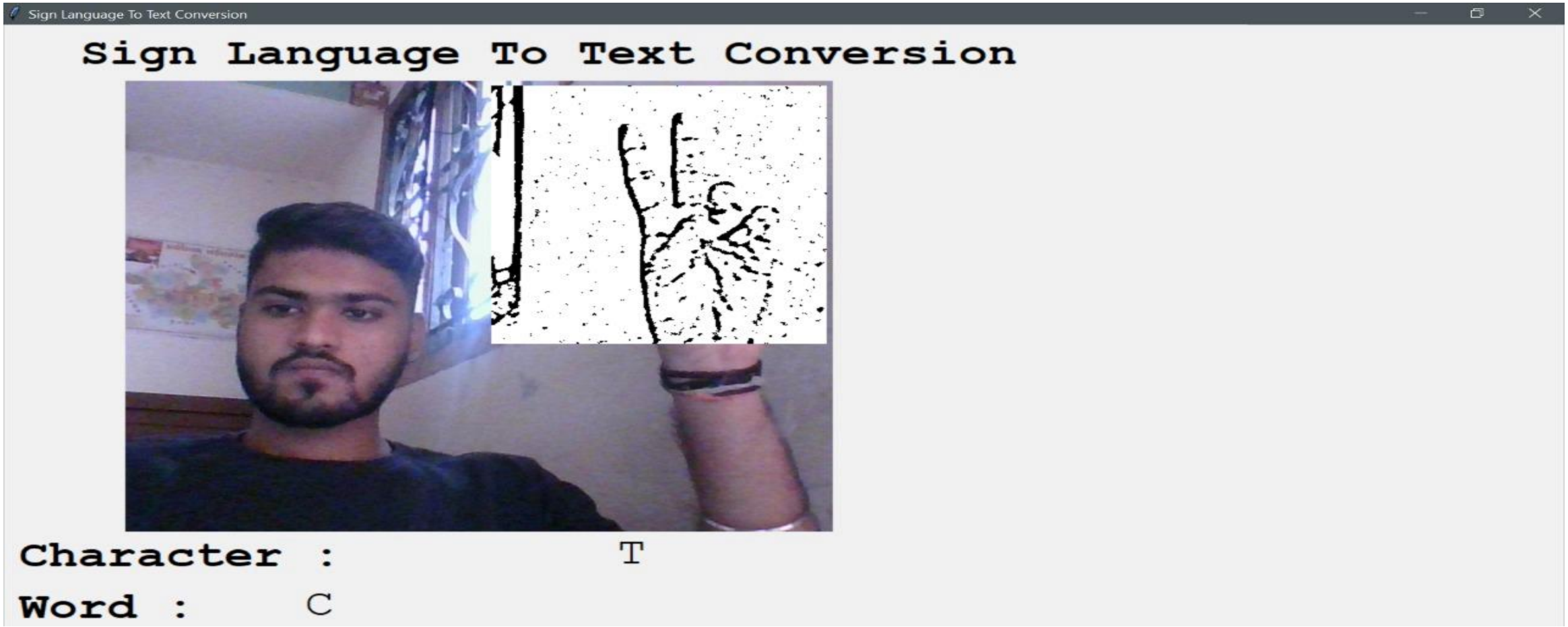
- **Steps of building this project:**

1. The first Step of building this project was of creating the folders for storing the training and testing data. As, in this project we have built our own dataset.
2. The second step, after the folder creation is of creating the training and testing dataset.
3. We captured each frame shown by the webcam of our machine .

Solution Proposed

4. In each frame We defined a region of interest (ROI) which is denoted by a blue bounded square .
5. After capturing the image from the ROI, we applied gaussian blur filter to the image(to mute noises) which helps for extracting various features of the image.
6. The image after applying gaussian blur looks like below.

Solution Proposed



Solution Proposed

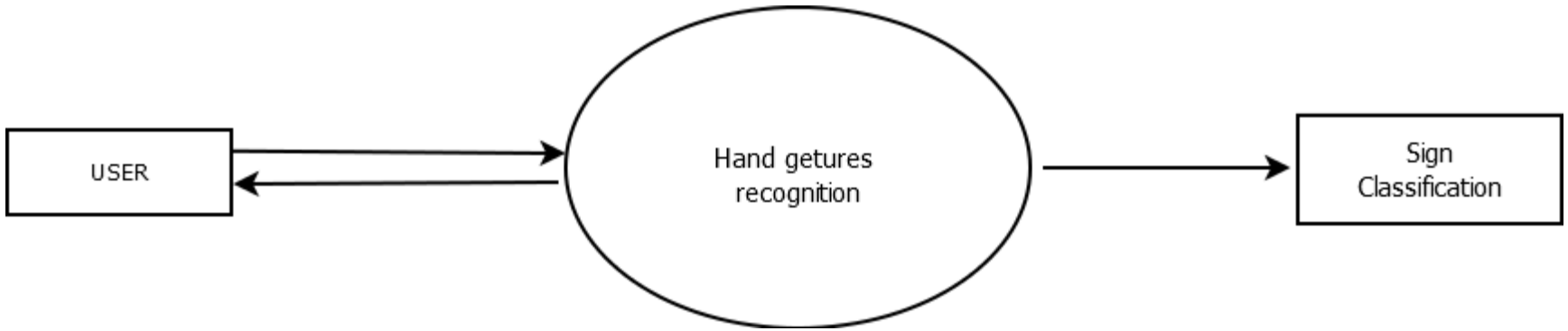
- After the creation of the training and testing data. The third step is of creating a model for training. Here, We have used Convolutional Neural Network(CNN) for building this model(CNN is a deep learning neural network sketched for processing structured arrays of data.)

- The final step after the model has been trained is of creating a GUI that will be used to convert Signs into text and form sentence, which would be helpful for communicating with D&M people.

Models/Diagrams

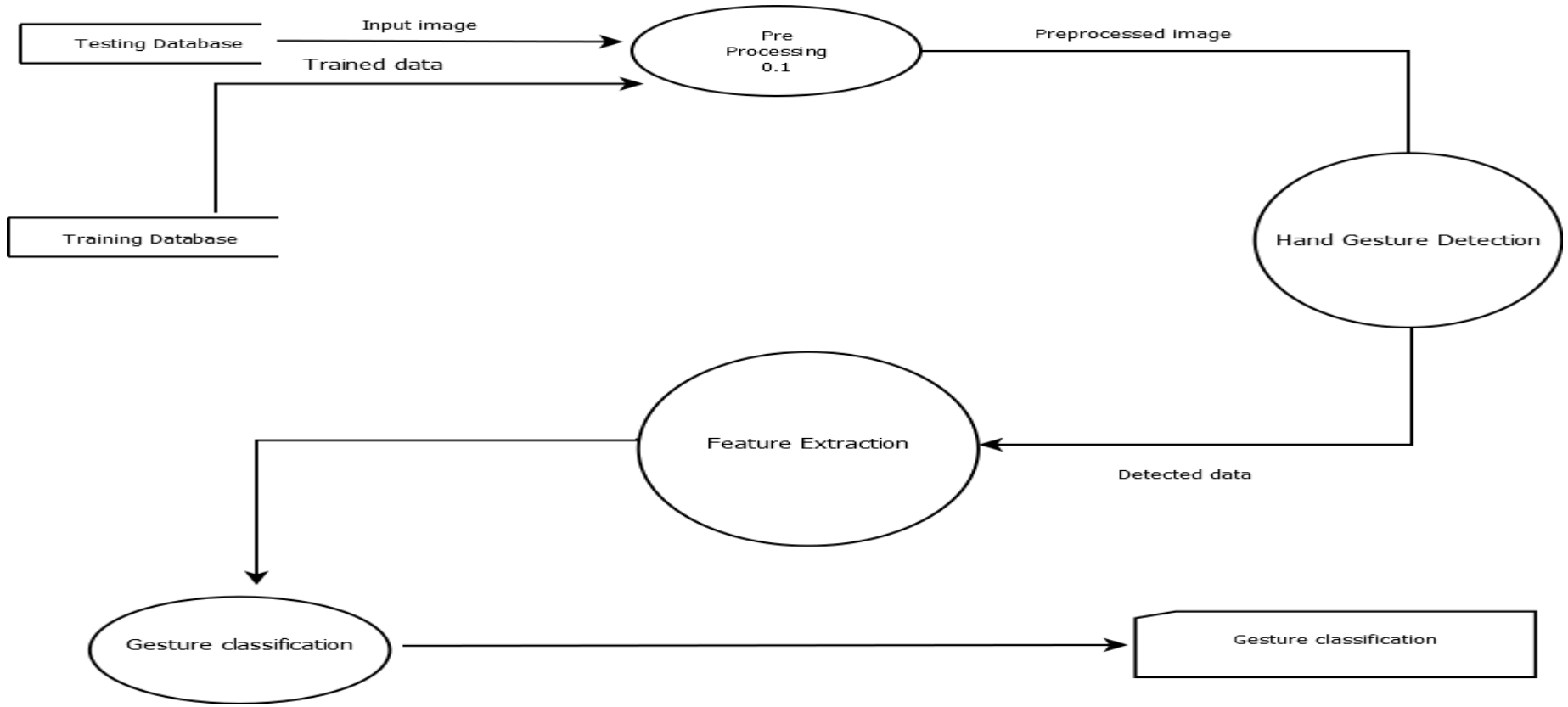
- Software Development Models
- System Architecture
- ER Diagram
- UML Diagrams
- DFD
- Flow Diagram

DFD



Level 0

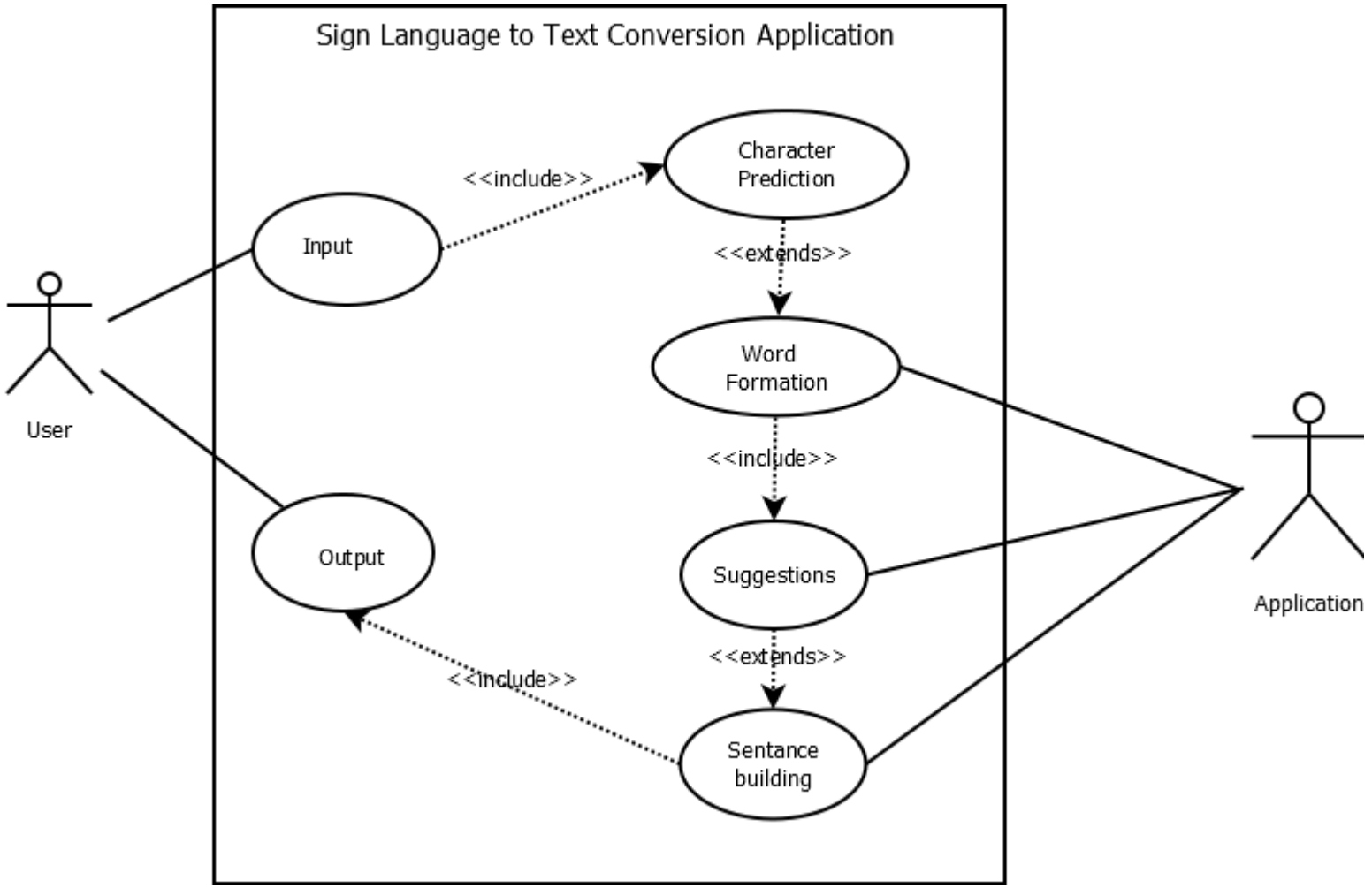
DFD



Level 1

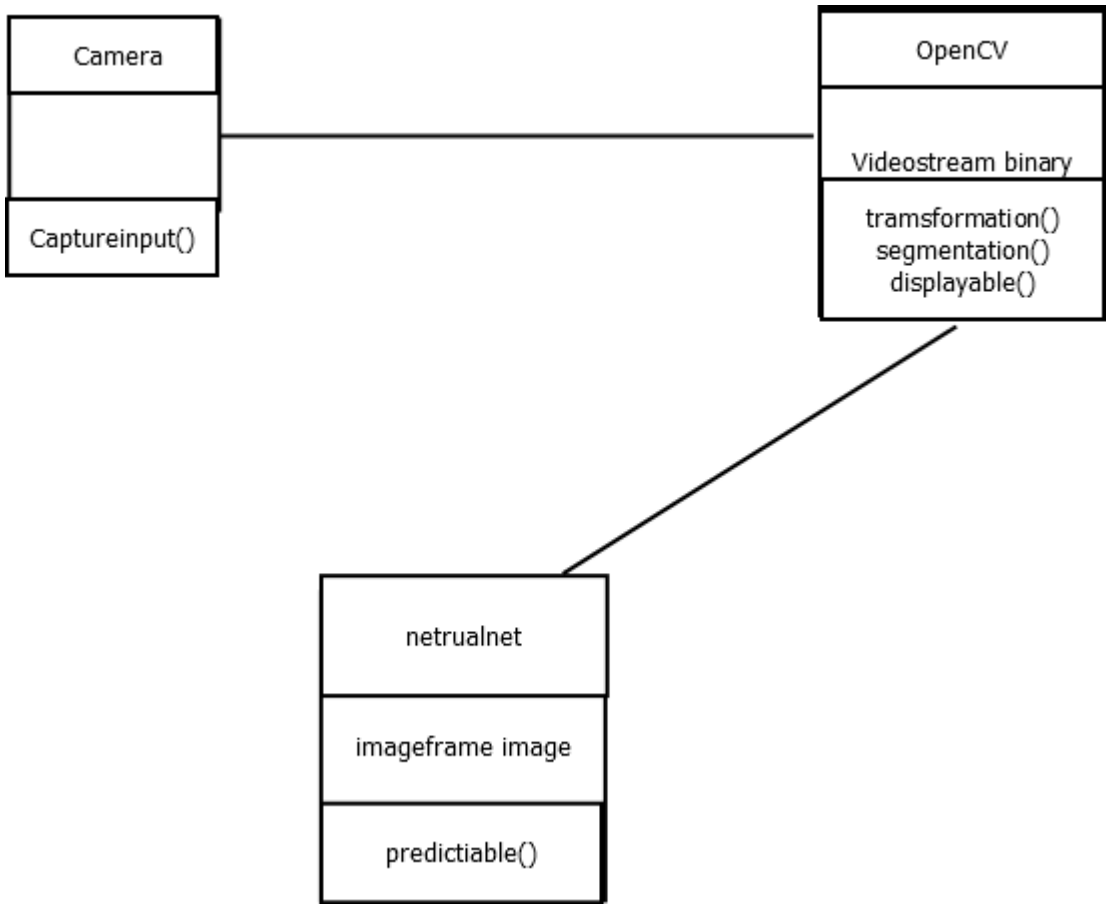
UML diagrams

- Use case:



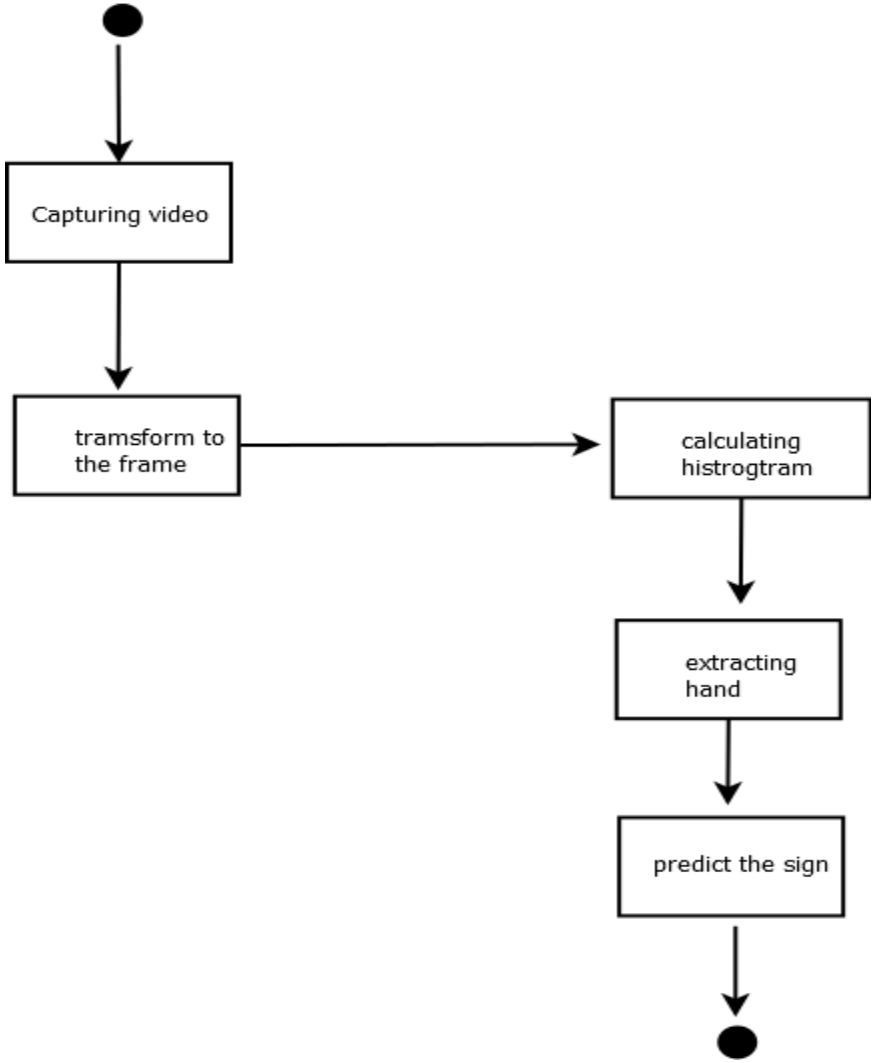
UML diagrams

- Class:



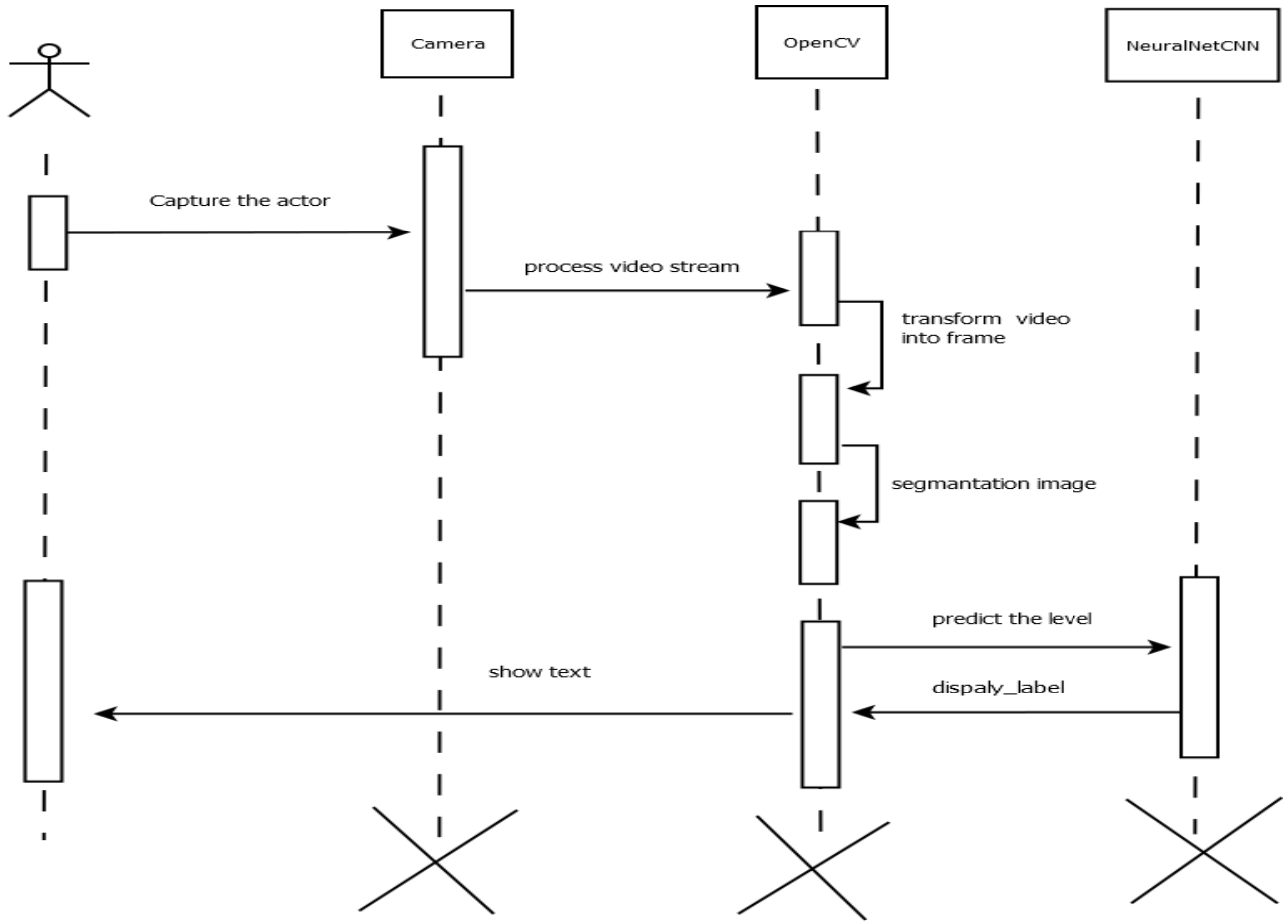
UML diagrams

- State Chart

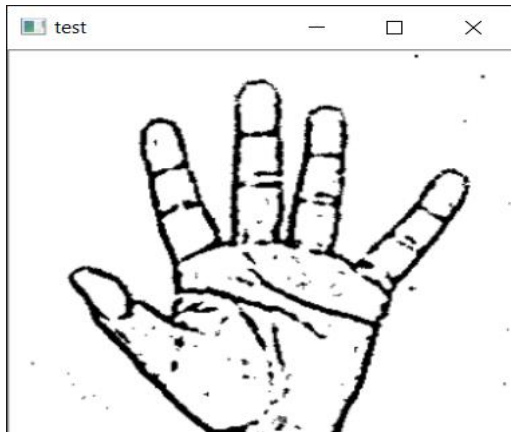


UML diagrams

- Sequence



Implementation



Layer 1

Classify between 27 Symbols

Layer 2

Classify between Similar Symbols

Implementation

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.

Implementation

- **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

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).

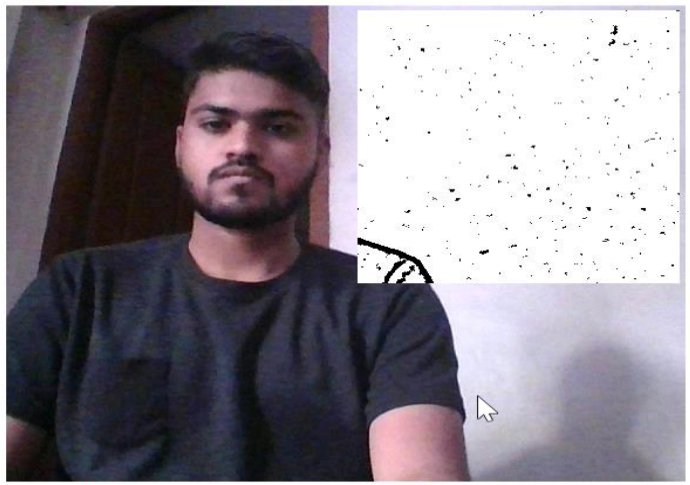
2. Otherwise we clear the current dictionary which has the count of detections of present symbol to avoid the probability of a wrong letter getting predicted.

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.

Implementation

Sign Language To Text Conversion



Character : G

Word : HTCL

Implementation

1. Lastest pip -> pip install --upgrade pip
2. numpy -> pip install numpy
3. string -> pip install strings
4. os-sys -> pip install os-sys
5. opencv -> pip install opencv-python
6. tensorFlow -> pip install tensorflow
7. keras -> pip install keras
8. tkinter -> pip install tk
9. PIL -> pip install Pillow
10. enchant -> pip install pyenchant (Python bindings for the Enchant spellchecking system)
11. hunspell -> pip install cyhunspell (A wrapper on hunspell for use in Python)

Conclusion

- We achieved final accuracy of **90.0%** on our data set. We have improved our prediction after implementing two layers of algorithms wherein we have verified and predicted symbols which are more similar to each other.
- This gives us the ability to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate.

Limitations

- The model works well only in good lighting conditions.
- Plain background is needed for the model to detect with accuracy.

Acknowledgement

First and foremost ,we would like to express our sincere gratitude to our mentor Krupi mam for the continuous support, guidance and her immense knowledge. Finally we would like to thank our team mates for all the support and encouragement during the project work.

Q&A

THANKS