**CHAPTER 1**

**INTRODUCTION** **AND PROBLEM STATEMENT**

* 1. **PROJECT SUMMARY**
* The world is hardly live without communication, no matter whether it is in form of textual, voice or visual expression.
* The communication among the deaf and dumb people is carried by text or visual expression.
* Gesture communication is always in the scope of confidential and secure communication.
* Hands and facial parts are immensely influential to express Face detection and the thoughts of human in confidential communication.
* Sign language is learned by deaf and dumb, and usually it is not known to normal people, so it becomes a challenge for communication between a normal and hearing impaired person.
* Its strike to our mind to bridge the between hearing impaired and normal people to make the communication easier.
* Sign language recognition (SLR) system takes an input expression from the hearing impaired person gives output to the normal person in the form text or voice.
  1. **OBJECTIVE**
* Communication is always having a great impact in every domain and how it is considered the meaning of the thoughts and expressions that attract the researchers to bridge this gap for every living being.
* The objective of this project is to identify the symbolic expression through images so that the communication gap between a normal and hearing impaired person can be easily bridged.
  1. **PROBLEM STATEMENT**
* To Identify the Hand Signs Gestures or Symbol from an Image to Recognition the Alphabets.

**CHAPTER 2**

**Tools, Technologies and Principles used**

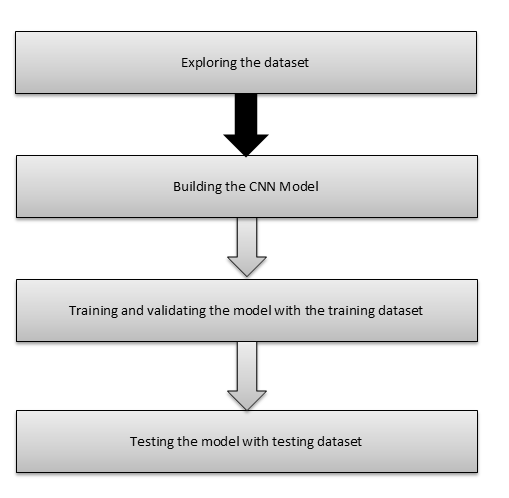
**2.1 Tools, Technologies and Principles used**

1. **Sign Language Recognition using Python Programming Language**.
2. **Application :** Console Application
3. **Required Software**
   1. VS Code
   2. Jupyter Notebook
4. **Required Libraries:**
   1. Pandas
   2. Numpy
   3. Matplotlib
   4. OpenCV
   5. Imutils
   6. PIL
   7. Tenserflow
   8. Keras
5. **Hardware Specification**
   1. Intel Core I3 Processor 1.6 GHz
   2. RAM : 4 Gigabyte
   3. ROM : 480 Gigabyte
6. **Operation System**
   1. Windows 10 Pro

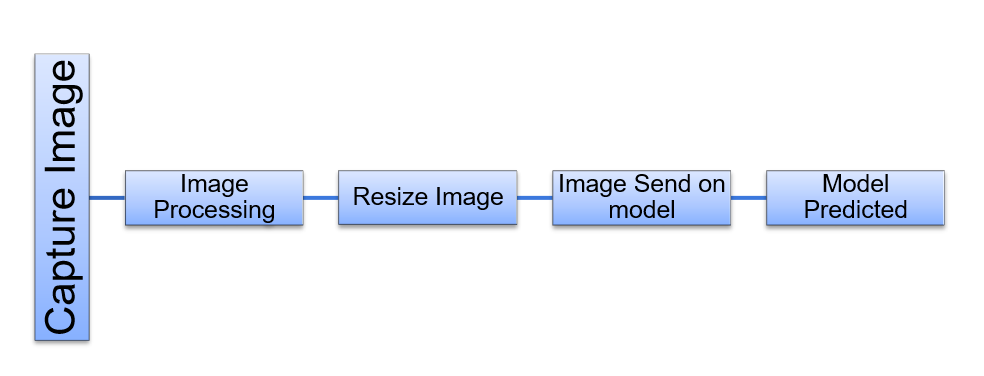
**CHAPTER 3**

**METHODOLOGY**

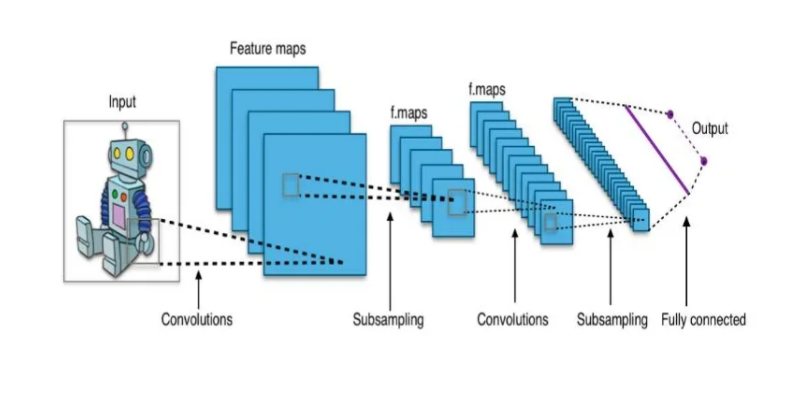
**3.1 TRAINING PROCESS**

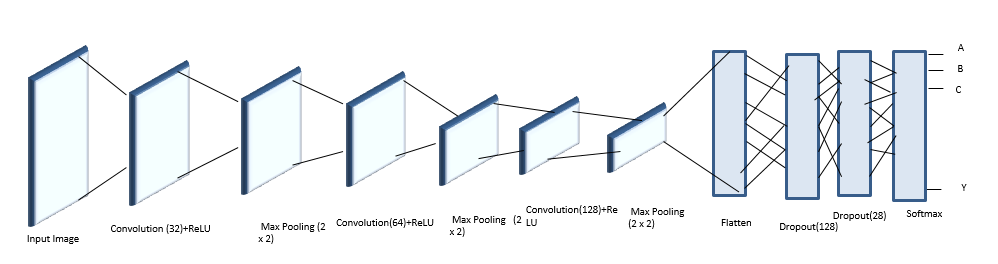
****

**3.2 DETECTION PROCESS**

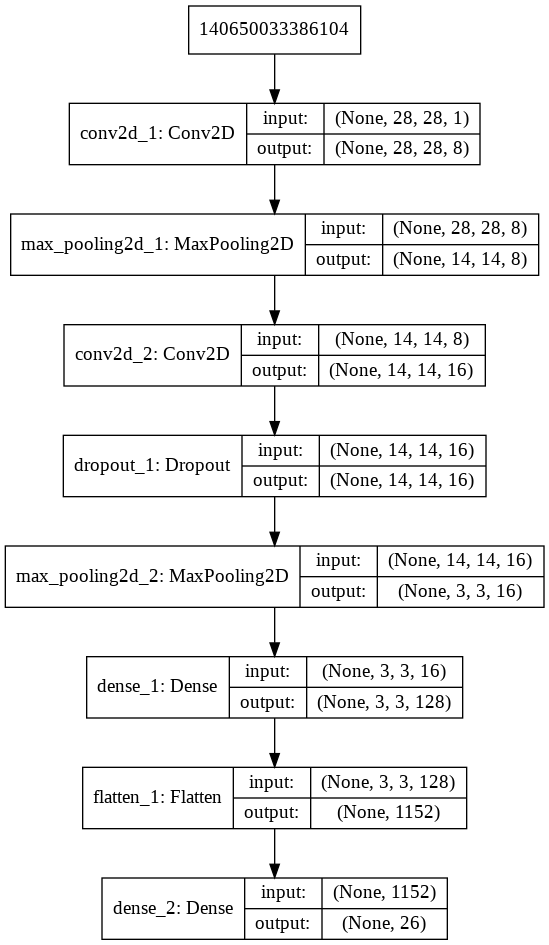
****

**3.3 CNN Model Architecture**

****

****

**3.4 CNN Model Architecture with Input and Output**

****

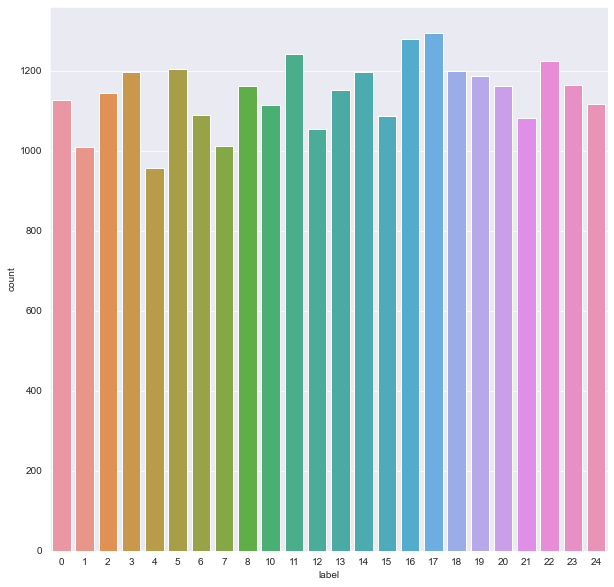
**CHAPTER 4**

**DATASET**

**4.1 Data Set that is used for Training the Model**

****

**4.2 Dataset Exploration**

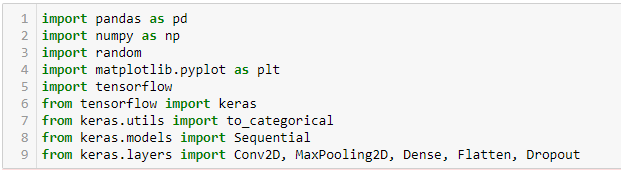
****

* The dataset format is patterned to match closely with the classic MNIST.
* Each training and test case marks a label from 0 to 25 as a matched map for every alphabetic letter A-Z (and no cases for 9=J or 25=Z due to gesture motions).
* Each class label is a set of sign images of the English alphabet.
* The training data (27,455 samples) and test data (7172 samples) are approximately half the size of the standard MNIST but otherwise similar with a header row of the label, pixel1, pixel2…. pixel784 which represents a single 28x28 pixel image with grayscale values ranging 0-255.
* The dataset seems balanced as for each training label, and enough training examples exist shown in Figure on the left side that we obtained during the exploration of the dataset.

**CHAPTER 5**

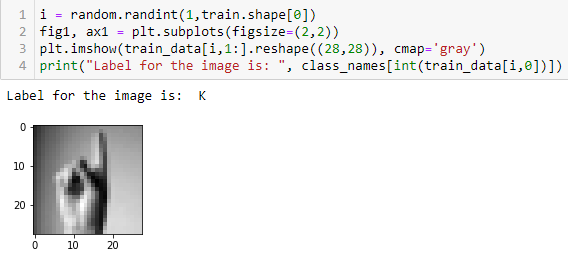
**IMPLEMENTATION DETAILS**

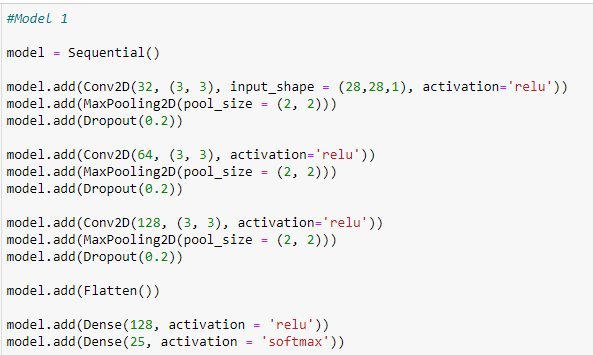
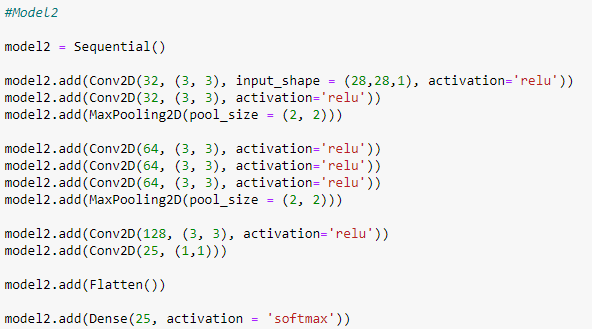
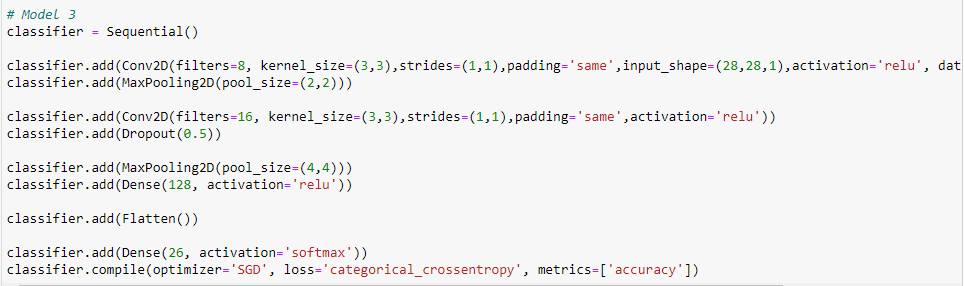
**5.1 IMPLEMENTATION DETAILS**

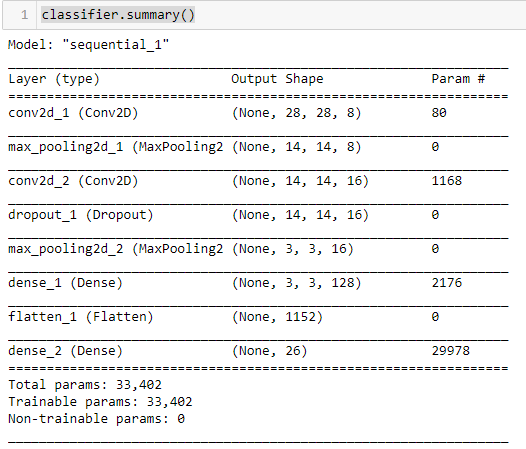
1. **Required Libraries:**
2. **Load Train and Test Data:**

****

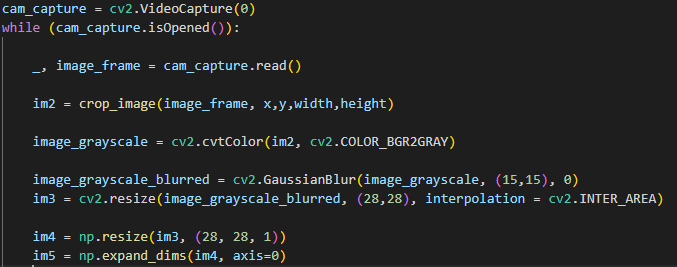
1. **Plot 1 random image from Training Data:**

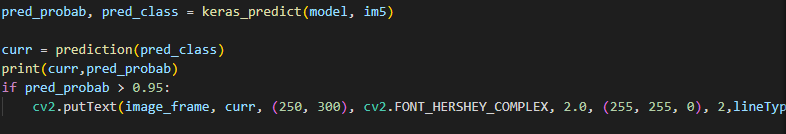
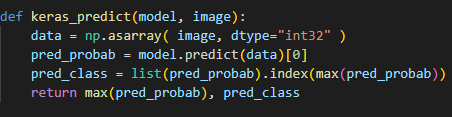
****

1.  **Normalize / Scale Data Set Values**
2. **Defining the Convolutional Neural Network (CNN) Model 1**
3. **Defining the Convolutional Neural Network (CNN) Model 2**
4. **Defining the Convolutional Neural Network (CNN) Model 3**
5. **Model Summary**

****

1. **Capture Video and Image Processing**



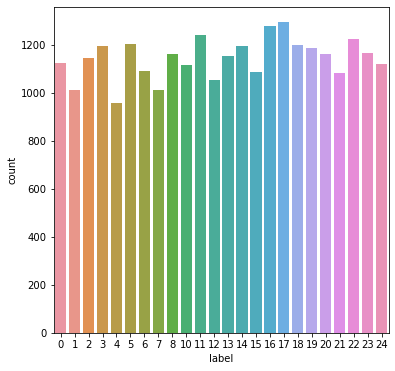
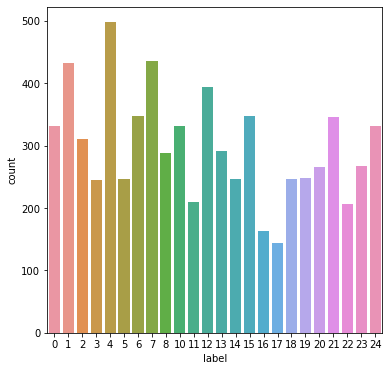
1. **Predicted the Class of a Sing Symbole**
2. **Predicted Function to predict the Class**

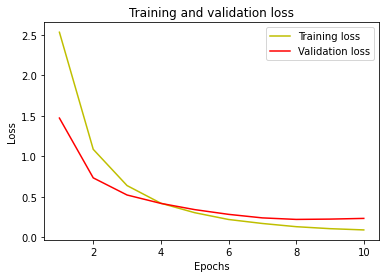
**CHAPTER 6**

**RESULT**

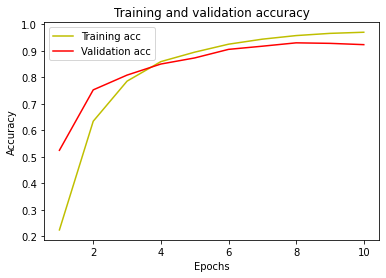
1. **Data Set Images**

****

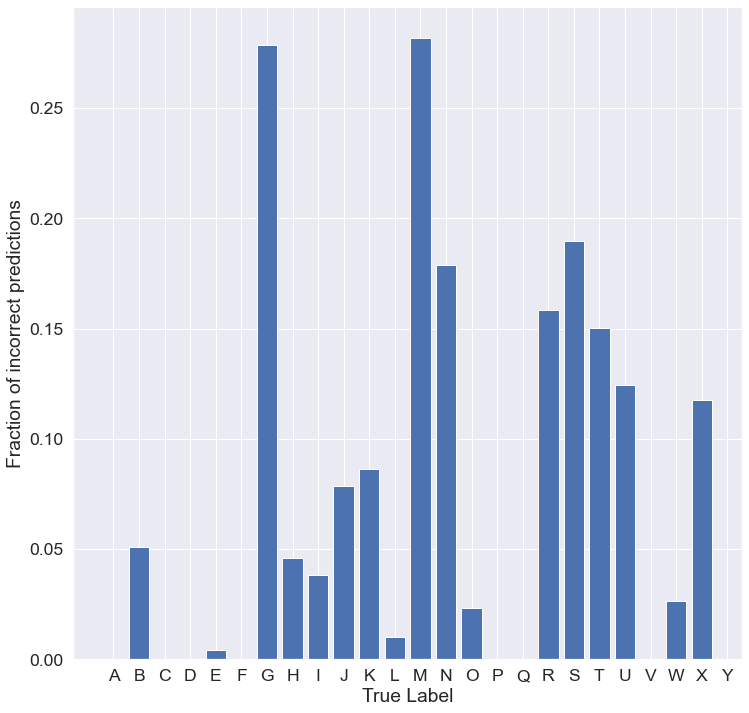
1. **Train Data Distribution Visualization**
2. ** Test Data Distribution Visualization**
3. **Training and validation loss**

****

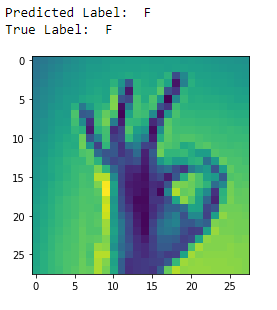
1. **Training and validation accuracy**

****

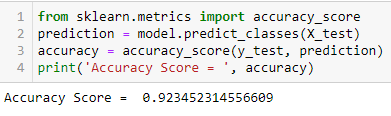
1. **Plot fractional incorrect misclassifications**

****

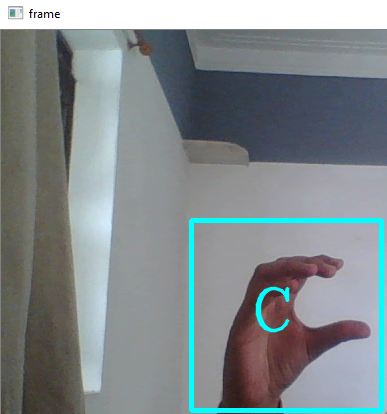
1. **Predict On Testing Data**

****

1. **Accuracy Score**

****

1. **Predict Using Camera**

****

**CHAPTER 7**

**CONCLUSION**

* Here, I am using Deep learning and computer vision techniques to classify Sign Language.
* I am using python programming and its different libraries for Deep learning and computer vision.
* My system has extract different feature of the Sign Languages so based on that features we can easily classify Alphabet.

**CHAPTER 8**

**FUTURE ENHANCEMENT**

* Future study may extend our work to accept video frames to include letters J and Z in the classification so that more varied inputs can be processed and understood by the network.
* Furthermore, there is a need for a large public dataset for automatic sign language recognition to utilize new deep learning techniques and a better way to benchmark performance

**CHAPTER 9**

**REFERENCES**

# References

|  |  |
| --- | --- |
| [1] | "Indian Sign Language Character Recognition," [Online]. Available: https://www.cse.iitk.ac.in/users/cs365/2015/\_submissions/vinsam/report.pdf. |
| [2] | "MIE324 Final Report: Sign Language Recognition," [Online]. Available: https://www.eecg.utoronto.ca/~jayar/mie324/asl.pdf. |
| [3] | tecperson, "Kaggle MINIST Data Set," [Online]. Available: https://www.kaggle.com/datasets/datamunge/sign-language-mnist. |
| [4] | M. Wurangian. [Online]. |