Real-time Hand Gesture Recognition for Motion Control in Blender

Name:	**Vishwanaathan B L**
Roll No:	20309
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE312

1 Introduction

Hand gesture recognition is a well known problem in recent years and there has been lot of developments on this topic. Here we try to merge with other applications. These kind of application can be seen used in augmented reality and video games like Pokemon Go. The main motive of the project is to recognize the particular gesture and the model in blender should move as it was progeammed for that particular gesture or posture. .

2 Proposed Dataset

Initial phase of this project I tried using MNIST dataset (signlanguage model) but it ended up having very low interclass variation leading to very bad accuracy. So we develop our own dataset only with 3 classes, Figure 1. A initial boundary of 170 X 170 is defined and then first initial frame is taken whenever we vring a object within that frame we take that image ans subtract from the first frame we stored. This leaves us with the isolated object which is latter converted to binary image.

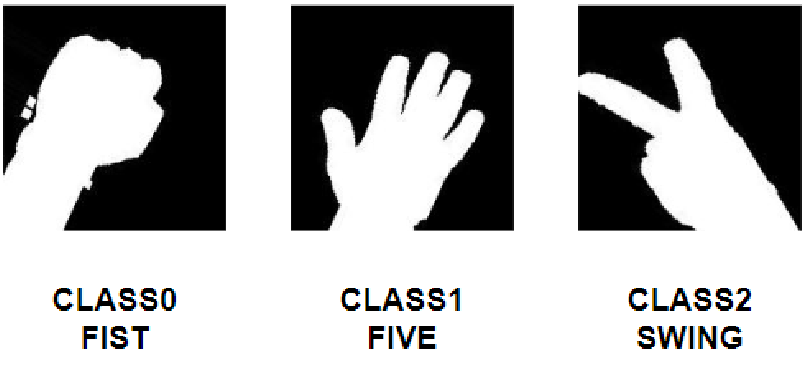

Figure 1: Classes used for training

The above collected images go through various image augmentations like horizontal shift, vertical shift, horizontal flip, zoom in, shear, rotation etc thereby producing a nice amount dataset as follows: (**training data: 2592, validation data: 897, test data: 897**).

3 Experiment and Results

To perform image classifcation and recognition we use diffferent librarires like **TensorFlow, OpenCV, Scikit-Learn**. The dataset is used to train the custom designed **CNN** model. Our model has 3 convolutional layer with 3 X 3 filter , 2 X 2 pool size, with activation function ReLU and 2 connected layer

Table 1: Metrics Of The Trained Model

Classes	Precision	Recall	F-measure
Class0	0.98	1.00	0.99
Class1	1.00	1.00	1.00
Class2	1.00	0.98	0.99

with a flattend layer connecting to 128 units and then connected to 3 units (which is the final 3 class classification) then an final activation layer using softmax function (since it is a multiclass problem).

The model compiled is with 'categorical crossentropy loss' function and 'adam' as optimizer. The model is then trained with steps per epoch of 2500, batch size 16 and 50 epochs in total . Table1 and Table2 shows the result of the CNN model trained and tested.

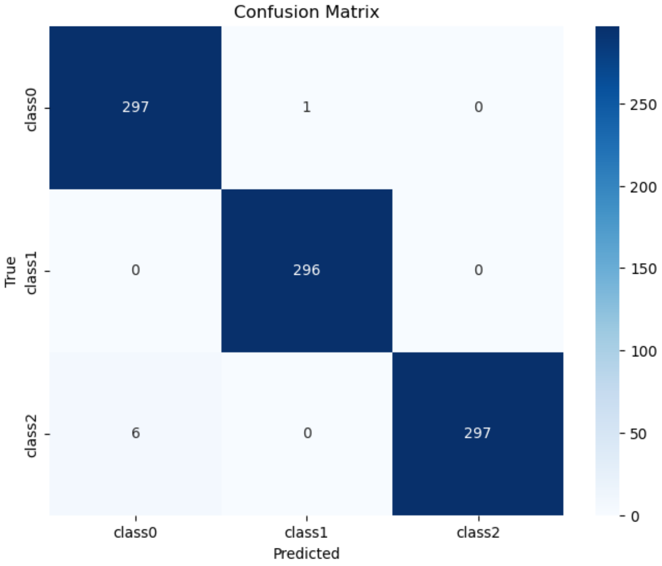

Figure 2: Confusion Matrix

4 Discussions and Conclusion

From the above results it can be seen that the model performs well on after testing it with data. The model is then saved and used with real time frame capture where each frame is passed through the model after being preprocessed as mentioned earlier. Then we use **pyserial** library here to communicate with applications like BLENDER. We add an algorithm upon if Class0 is been detected pause the movement, Class1 is detected increment the location of object (Y axis) and if Class2 is detected decrement the location of the object (Y axis).

Our model has certain limitations like tilting of the gesture wont be recognized, there shouldn't be any other object moving in that defined frame and we also loose the depth information here while converting the images to binary. The changes and improvements can be made in the model further on where we can even track the movement of the arm and replicate it in he model. The output video is made available in the references.

5 References

- Used to refer Binary Image Classification: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8321080/
- https://www.tensorflow.org/tutorials/images/cnn
- sklearn.model_selection.GridSearchCV
- Link To The Output