MODELLING

Part 1

Group Members

Shrey Raval - 100730265

Jigar Patel - 100730261

Nikhil Singh - 100766644

Tarunjot Singh - 100766653

Model Structure

To detect sign Language, we have used two models CNN and LeNet-5. Both of the models have their own structure and procedure for data fitting and testing. Model Structure is Discussed below.

CNN:

A Convolutional Neural Network (CNN) is a deep learning algorithm that can identify and distinguish features in computer vision videos. This is a multi-layer neural network built to process visual signals and perform tasks such as image recognition, segmentation and object detection.

Architecture

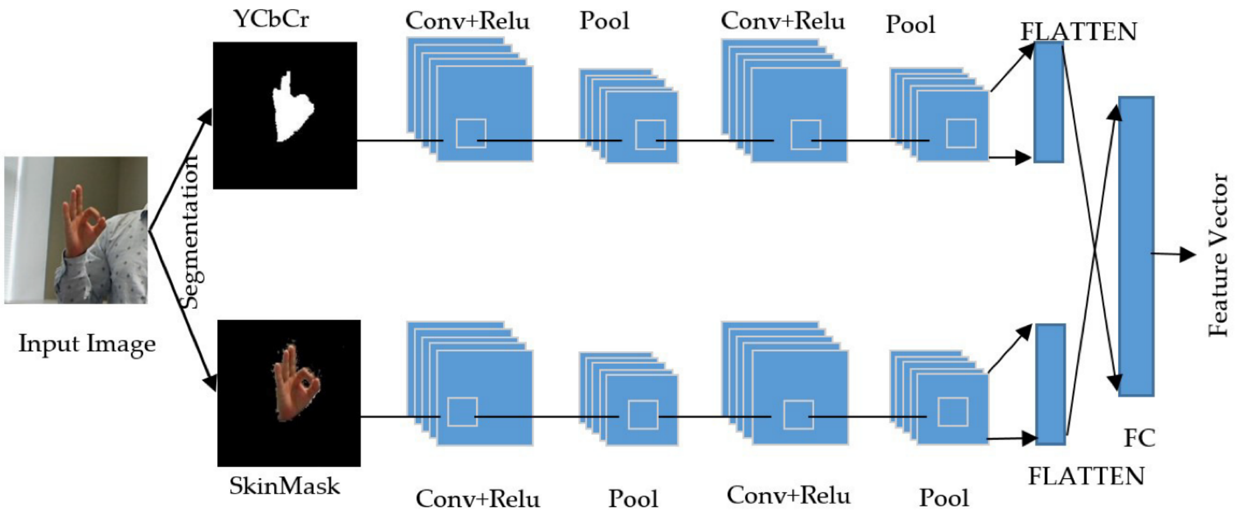
CNN's architecture is influenced by the structure and flexibility of the visual cortex and is designed to mimic the process of neuron interaction within the human brain.

The neurons inside CNN are divided into a three-dimensional structure, with each group of neurons examining a small region or function of the picture. CNNs use layer projections to generate a final output that provides a probability score vector to reflect the possibility that a given function belongs to a certain class.

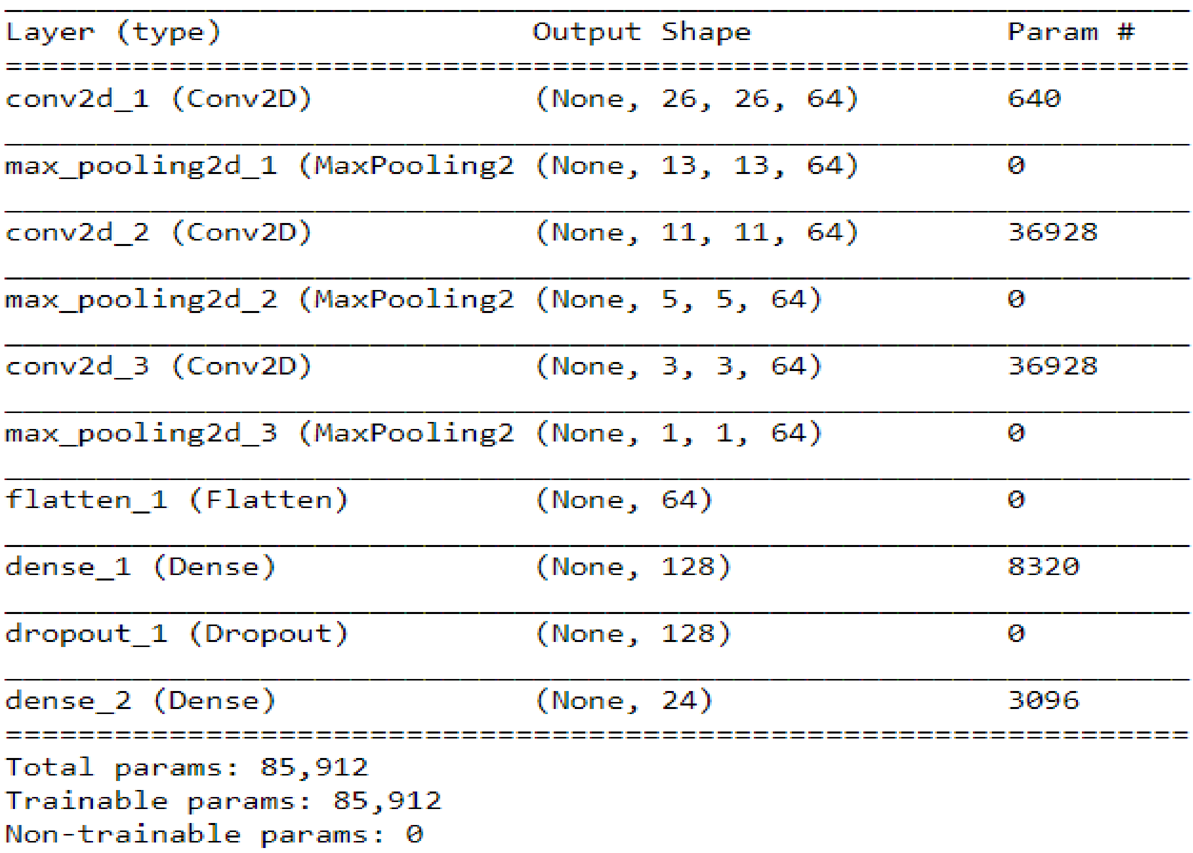
CNN is made up of several types of bricks.

Convolutional layer: It creates a function map to predict the class probabilities for each element by adding a philtre that scans the entire image, few pixels at a time**.**

Pooling layer: It helps to Scale down the amount of knowledge produced by the convolutionary layer for each function and preserving the most essential information.



Detail summary of the Model Layers:



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 64)        640       
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 64)        0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 11, 11, 64)        36928     
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 64)          0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 3, 3, 64)          36928     
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 1, 1, 64)          0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 64)                0         
_________________________________________________________________
dense_1 (Dense)              (None, 128)               8320      
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0         
_________________________________________________________________
dense_2 (Dense)              (None, 24)                3096      
=================================================================
Total params: 85,912
Trainable params: 85,912
Non-trainable params: 0
```

Model includes 3 levels of Convolution layers with the output shape of (26, 26, 64) and total number of input parameters are 640. Further we have used max pooling to reduce the dimension of the image dataset which result in low computational cost for further processing. On top of that this layer also helps to extract the important piece of information from images. Moving further, this model has the second convolution layer followed by the third convolution layers. Coming towards the end there is a dropout feature which help to overcome the problem of model overfitting and then output is obtained.

LeNET-5

Lenet-5 is a type of simple CNN which consists of 7 Layers. Out of these 7 Layers, 3 are the convolutional layers, 2 are pooling layers (sub-sampling), 1 Fully connected and lastly 1 output layer. We will be implementing the Le-net 5 architecture because the le-net architecture was initially developed for the recognition of handwritten digits and alphabets and is used by majority of U.S banks till Date to read cheques.

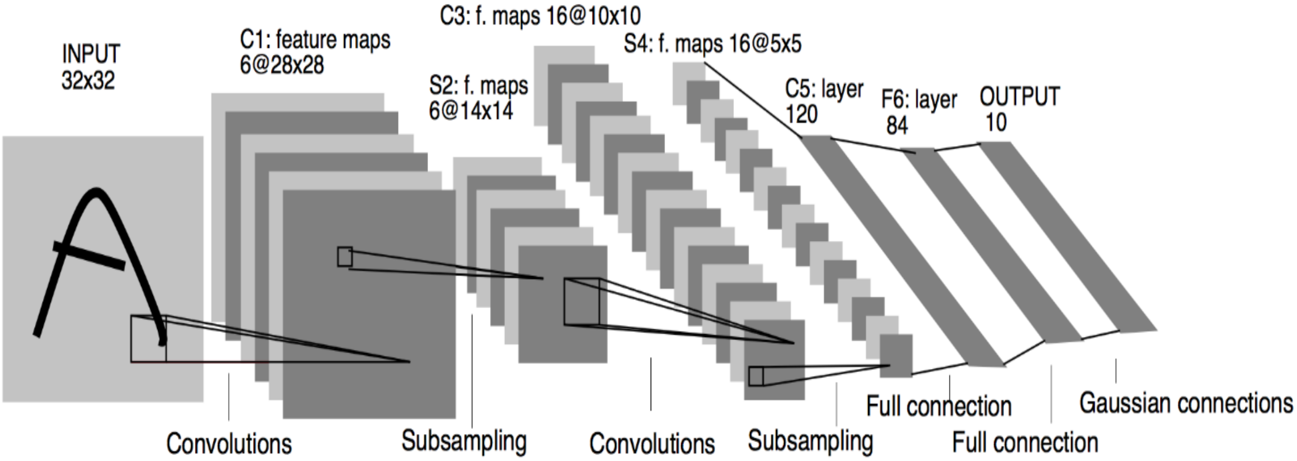
Following are the specifications of the layers:

Convolutional layer will be using the 5x5 convolutions with stride.

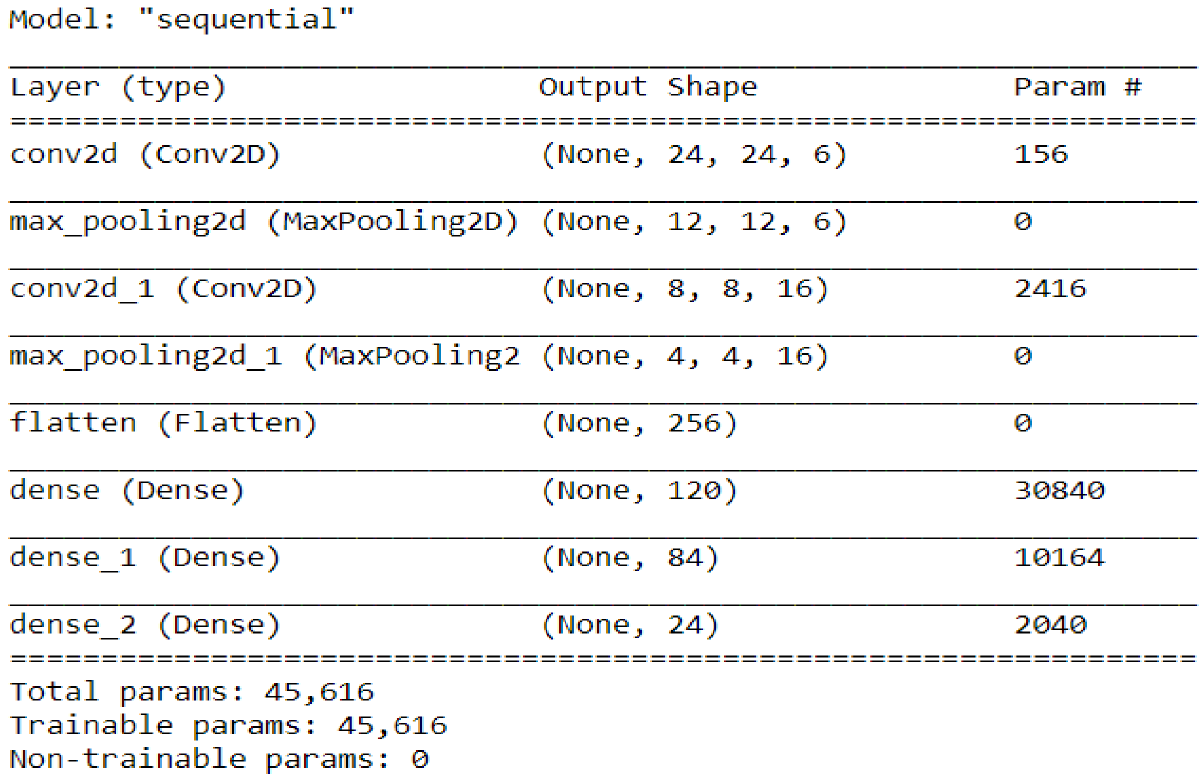
Pooling Layer will consist of 2x2 Average Pooling

Activation Function throughout the entire network will be consisting of only tan h and sigmoid.

During the research regarding various architecture we came to know that as LeNet works with digit and character recognition in a great way it would also work in a similar way for our problem also as our scope includes the recognition of hand gestures.



Detailed Summary of Model Layers:



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 24, 24, 6)         156
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 6)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 8, 8, 16)          2416
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 4, 4, 16)          0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 120)               30840
_________________________________________________________________
dense_1 (Dense)              (None, 84)                10164
_________________________________________________________________
dense_2 (Dense)              (None, 24)                2040
=================================================================
Total params: 45,616
Trainable params: 45,616
Non-trainable params: 0
```

Model includes 2 levels of Convolution layers out of which the 1st has the output shape of (24, 24, 6) and the 2nd one has and output shape of (8, 8, 16) and total number of input parameters are 156 and 2416 respectively. Further we have used max pooling that reduces the dimension of the image dataset which results into saving a lot of computation power as well as processing time. Lastly, we have the dense layer which will act as a regular layer of neurons within the network.

Software Tools:

For this project we have used the following software and tools:

Python as the Programming language.

Jupyter Notebook for the IDE purpose.

Atom Text Editor.

Nvdia Cuda to train model.

Python flask to create end point.

JavaScript for the creating Front end view.

OpenCv for live image Processing.

References:

1. https://www.azuredevopslabs.com/labs/vstsextend/aml/
2. https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac
3. http://deeplearning.net/tutorial/lenet.html
4. https://medium.com/@pechyonkin/key-deep-learning-architectures-lenet-5-6fc3c59e6f4