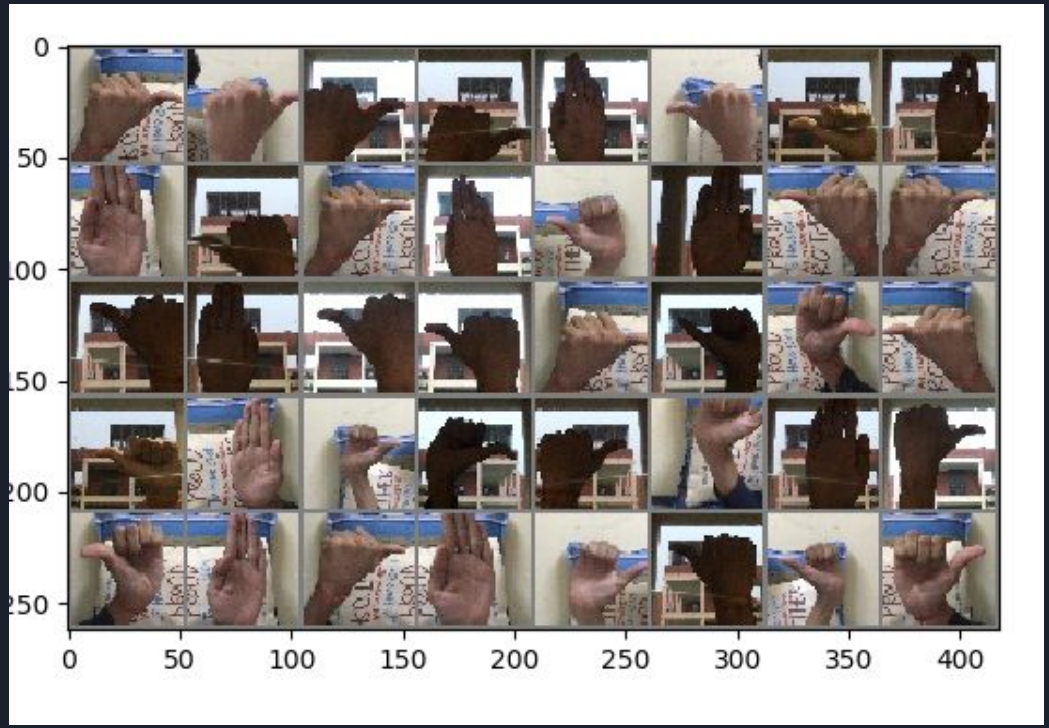
Hand Gesture Recognition

Dataset Preparation - I

1. Images for each gesture were taken in multiple in multiple pose, lighting, background, scale etc.
2. Dataprepare.py script takes an video input and segrates it the video into datasets folder. We created 3 datasets for each gesture : Test, Training and Cross-Validation.
3. Training Class consisting of nearly 3,100 images for each class whereas cross-validation and test dataset consists of 400 and 200 images respectively.
4. Script consists of functions to resize, crop, add text , add rectangle etc.
5. Various modifications are done on the data such as background subtraction, edge detection, grayscale etc.

Dataset Preparation - II

https://drive.google.com/file/d/1yEdbo3AuS2ug3S9-daNMGr_FZ-HZFHrv/view?usp=sharing (Link)

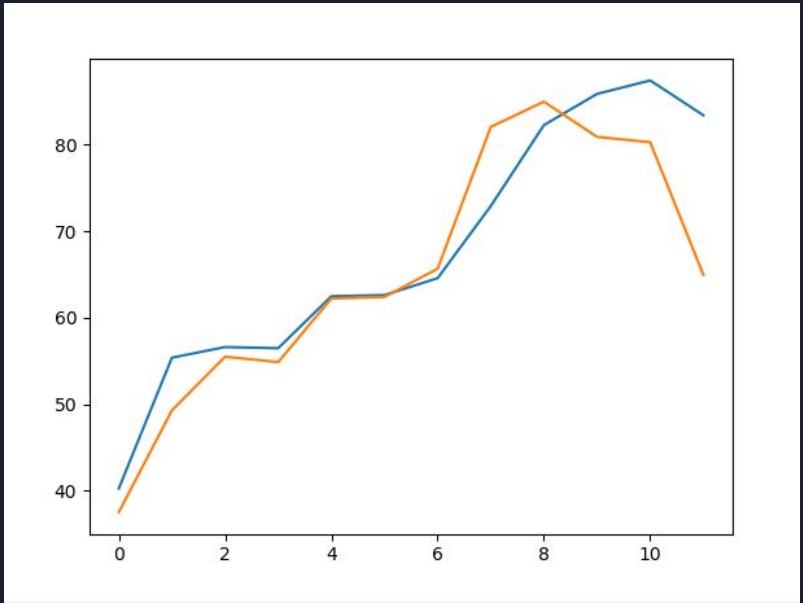


CNN Architecture

```
class Net(nn.Module):
  def __init__(self):
    super(Net, self).__init__()
    self.conv1 = nn.Conv2d(3, 12, 3,padding = 0)
    self.pool = nn.MaxPool2d(2, 2)
    self.conv2 = nn.Conv2d(12, 30, 3,padding = 1)
    self.pool = nn.MaxPool2d(2,2)
    self.conv3 = nn.Conv2d(30,40,3)
    self.pool = nn.MaxPool2d(2,2)
    self.fc1 = nn.Linear(40*5*5,100)
    self.fc2 = nn.Linear(100,3)

  def forward(self, x):
    x = self.pool(F.relu(self.conv1(x)))
    x = self.pool(F.relu(self.conv2(x)))
    x = self.pool(F.relu(self.conv3(x)))
    x = x.view(-1, 40*5*5)
    x = F.relu(self.fc1(x))
    x = (self.fc2(x))
    # x = self.fc3(x)
    return x
```

Graph Accuracy:
Training, Cross-Validation vs Epoch



1. The blue line denotes the training accuracy while the orange line denotes the cross-validation accuracy.
2. We can see that cross -validation accuracy and training accuracy are both high after epoch 7.
3. After this point, data is being overfitted into the model.
4. Model state after epoch 7 is supposed to give best accuracy on test dataset.

Optimization / HyperParameter Tuning

1. Batch-Gradient Descent was used with mini-batch of size 32 as it gave better results compared to batches of size 16,64,24,48
2. Total of 16 epochs were run:
    a. For the first 8 epoch, learning rate is 0.001 and momentum 0.9
    b. In the next 4 epoch, learning rate is 0.0005 and momentum 0.6
    c. In the next 4 epoch, learning rate and momentum is further reduced to fine tune the CNN Model.
3. Contropy Loss is used as the criteria for optimizing the model.
4. Threshold for the background class was tuned by testing.

Pre-processing Frame

1. Algorithm to get the mask:
    a. Background subtraction:
        i. Either KNN or MOG2
    b. Skin Detection using a pixel based decision tree:
        i. Combination of skin detection and background subtraction
2. The mask is used to get contours and identify bounding box
    a. For each box the inference is made, if it exceeds a threshold then either left, right or stop is predicted, otherwise STOP is predicted.