

BUILDING A FLASK APPLICATION PART-3

Code to preprocess the frame captured from camera :

```
def detect(frame):  
    img=resize(frame,(64,64,1))  
    img=np.expand_dims(img,axis=0)  
    if(np.max(img)>1):  
        img=img/255.0  
    with graph.as_default():  
        prediction = model.predict_classes(img)  
    print(prediction)  
    pred=vals[prediction[0]]
```

camera.py

```
import cv2  
import numpy as np  
from tensorflow.keras.models import load_model  
from tensorflow.keras.preprocessing import image  
  
class Video(object):  
    def __init__(self):  
        self.video = cv2.VideoCapture(0)  
        self.roi_start = (50, 150)  
        self.roi_end = (250, 350)  
        self.model = load_model('asl_model.h5') # Execute Local Trained Model  
        # self.model = load_model('IBM_Communication_Model.h5') # Execute IBM  
Trained Model  
        self.index=['A','B','C','D','E','F','G','H','I']  
        self.y = None  
    def __del__(self):  
        self.video.release()  
    def get_frame(self):  
        ret,frame = self.video.read()  
        frame = cv2.resize(frame, (640, 480))  
        copy = frame.copy()
```

```
copy = copy[150:150+200,50:50+200]
# Prediction Start
cv2.imwrite('image.jpg',copy)
copy_img = image.load_img('image.jpg', target_size=(64,64))
x = image.img_to_array(copy_img)
x = np.expand_dims(x, axis=0)
pred = np.argmax(self.model.predict(x), axis=1)
self.y = pred[0]
cv2.putText(frame,'The Predicted Alphabet is:
'+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
ret,jpg = cv2.imencode('.jpg', frame)
return jpg.tobytes()
```