

```
from flask import Flask,render_template,request

import cv2

from keras.models import load_model

import numpy as np

from gtts import gTTS

import os

from keras.preprocessing import image

from skimage.transform import resize

from playsound import playsound

app = Flask(__name__)


model=load_model("aslpng1.h5")


vals = ['A', 'B','C','D','E','F','G','H','I']


@app.route('/', methods=['GET'])

def index():

    return render_template('index.html')

@app.route('/index', methods=['GET'])

def home():

    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])

def predict():

    print("[INFO] starting video stream...")

    vs = cv2.VideoCapture(0)


    (W, H) = (None, None)
```

```

while True:

    (grabbed, frame) = vs.read()

    if not grabbed:

        break

    if W is None or H is None:

        (H, W) = frame.shape[:2]

    output = frame.copy()

    # r = cv2.selectROI("Slect", output)

    # print(r)

    cv2.rectangle(output, (81, 79), (276,274), (0,255,0), 2)

    frame = frame[81:276, 79:274]

    frame = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)

    _, frame = cv2.threshold(frame, 95, 255,
cv2.THRESH_BINARY_INV)

    frame = cv2.cvtColor(frame, cv2.COLOR_GRAY2RGB)


    img = resize(frame,(64,64,3))

    img = np.expand_dims(img,axis=0)

    if(np.max(img)>1):

        img = img/255.0


    result = np.argmax(model.predict(img))

    index=['A', 'B','C','D','E','F','G','H','I']

    result=str(index[result])

```

```
        cv2.putText(output, "The Predicted Letter :  
{0}".format(result), (10, 50), cv2.FONT_HERSHEY_PLAIN,  
                    2, (150,0,150), 2)  
  
        cv2.putText(output, "Press q to exit", (10,450),  
cv2.FONT_HERSHEY_PLAIN, 2, (0,0,255), 2)  
  
        speech = gTTS(text = result, lang = 'en', slow = False)  
  
        cv2.imshow("Output", output)  
  
        key = cv2.waitKey(1) & 0xFF  
  
        if key == ord("q"):  
            break  
  
        print("[INFO] cleaning up...")  
        vs.release()  
        cv2.destroyAllWindows()  
        return render_template("index.html")  
  
if __name__ == '__main__':  
    app.run(debug=True)
```