Final Code

Make sure that you have installed the following packages:

- absl-py==1.0.0
- astunparse==1.6.3
- bidict==0.22.0
- cachetools==5.1.0
- certifi==2022.5.18.1
- charset-normalizer==2.0.12
- click==8.1.3
- colorama==0.4.4
- Flask==2.1.2
- Flask-SocketIO==5.2.0
- flatbuffers==1.12
- gast==0.4.0
- google-auth==2.6.6
- google-auth-oauthlib==0.4.6
- google-pasta==0.2.0
- grpcio==1.46.3
- h5py==3.7.0
- idna==3.3
- itsdangerous==2.1.2
- Jinja2==3.1.2
- keras==2.9.0
- Keras-Preprocessing==1.1.2

- libclang==14.0.1
- Markdown==3.3.7
- MarkupSafe==2.1.1
- numpy==1.22.4
- oauthlib==3.2.0
- opencv-python==4.5.5.64
- opt-einsum==3.3.0
- packaging==21.3
- Pillow==9.1.1
- protobuf==3.19.4
- pyasn1==0.4.8
- pyasn1-modules==0.2.8
- pyparsing==3.0.9
- python-engineio==4.3.2
- python-socketio==5.6.0
- requests==2.27.1
- requests-oauthlib==1.3.1
- rsa==4.8
- six==1.16.0
- tensorboard==2.9.0
- tensorboard-data-server==0.6.1
- tensorboard-plugin-wit==1.8.1
- tensorflow==2.9.1
- tensorflow-estimator==2.9.0
- tensorflow-io-gcs-filesystem==0.26.0
- termcolor==1.1.0

- typing_extensions==4.2.0
- urllib3==1.26.9
- Werkzeug==2.1.2
- wrapt==1.14.1

This application was developed and executed using Flask

Code:

```
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')
        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()
```