**AI ENABLED WEED RECOGNITION SYSTEM**

**A Mini - Project Report submitted in partial fulfilment of the requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

G RUTWIZ GANGADHAR

**Under the esteemed guidance of**

**M. Raja Mani**

**Assistant Professor, GIT**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM**

**(Deemed to be University)**

**VISAKHAPATNAM**

**JANUARY 20****21**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**



**DECLARATION**

We, hereby declare that the Project review entitled “**AI Enabled Weed Recognition System**” is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date: Jan 26th, 2021.

**Registration No: Name: Signature:**

**121710314020 G RUTWIZ GANGADHAR RUTWIZ**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM (Deemed to be University)**



**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled ​ **“AI Enabled Weed Recognition System”** is a bonafide record of work carried out by​G RUTWIZ GANGADHAR (121710314020),submitted in partial fulfilment of requirementfor the award of degree of Bachelor of Technology in Computer Science and Engineering.

**PROJECT GUIDE HEAD OF THE DEPARTMENT**

**M. Raja Mani Dr.K. THAMMI REDDY**

**(Asst. Professor) (PROFESSOR)**

**CSE, GIT CSE, GIT**

**GITAM GITAM**

**ACKNOWLEDGEMENT**

We would like to thank our project guide **Mrs. M. Raja Mani**, **Assistant. Professor**, Department of CSE for her stimulating guidance and profuse assistance. We shall always cherish our association with her for her guidance, encouragement and valuable suggestions throughout the progress of this work. We consider it a great privilege to work under her guidance and constant support

We also express our thanks to the project’s reviewers **Mrs. Bhramaramba Ravi**, Professor, and **Mrs. M Venkata Ramana**, **Assistant. Professor**, Department of CSE, GITAM (Deemed to be University) for their valuable suggestions and guidance for doing our project.

We consider it is a privilege to express our deepest gratitude to **Dr. K. Thammi Reddy**, Head of the Department, Computer Science Engineering for his valuable suggestions and constant motivation that greatly helped us to successfully complete this project.

Our sincere thanks to **Dr. C. Dharma Raj**, Principal, GITAM Institute of Technology, GITAM (Deemed to be University) for inspiring us to learn new technologies and tools.

Finally, we deem it a great pleasure to thank one and all that helped us directly and indirectly throughout this project.

**G RUTWIZ GANGADHAR 121710314020**

**TABLE OF CONTENTS**

**CONTENTS PAGE.NO**

|  |  |  |
| --- | --- | --- |
|  | **Declaration** | **i** |
|  | **Certificate** | **ii** |
|  | **Acknowledgement** | **iii** |
|  |  |  |
| **1.** | **Introduction** | **6** |
|  | 2.1 - Overview | 6 |
|  | 2.2 - Purpose | 6 |
| **2.** | **Literature Survey**  3.1 -Existing Problem  3.2 -Proposed Solution | **7**  **7**  **7** |
| **3.** | **Theoretical Analysis** | **8** |
|  | 3.1 – Block Diagram | 8 |
|  | 3.2 - Hardware/Software Designing | 9 |
| **4.** | **Experimental Investigations** | **10** |
| **5.** | **Flowcharts** | **11** |
| **6.** | **Results** | **12** |
| **7.**  **8.**  **9.**  **10.**  **11.**  **12.** | **Advantages and Disadvantages**  **Applications**  **Conclusion**  **Future Scope**  **References**  **Appendix**  12.1-Model Testing | **15**  **16**  **17**  **17**  **18**  **19**  19-29 |
|  |  |  |

**LIST OF FIGURES**

**Figure No. Description Page No.**

3.1 Block Diagram . 8

5.0 Flow chart 11

**LIST OF TABLES**

**Table No. Description Page No.**

7.0 Advantages and Disadvantages . 15

**1.Introduction**

**1.1 Overview**

Weeds are considered to be one of the biggest problems in agronomy. Their adverse effect is widely known—they reduce crop yields, serve as hosts for crop diseases and also produce toxic substances. Manual removal of weeds is labour-intensive, while the use of chemicals has long-term environmental consequences. Currently, approximately 23,000 tonnes of chemical herbicide at a cost of around £400million are used annually in the UK on weed prevention (Marchant, 1996). Reducing this quantity would potentially lead to reduced herbicide residues in water, food crops and the environment. One possible method of achieving this is to improve the selectivity with which herbicidal agents are applied to fields, by automated visual discrimination between crops and weeds for spot application of herbicide .Computer vision has been shown to provide a viable option in inspection of agricultural products, particularly when colour and shape need to be analysed at high speed (Batchelor and Searcy, 1989).

**1.2 Purpose**

Weed detection systems are important solutions to one of the existing agricultural problems—unmechanized weed control. Weed detection also helps provide a means of reducing or eliminating herbicide use, mitigating agricultural environmental and health impact, and improving sustainability.

**2. Literature Survey**

**2.1 Existing Problem**

Weed picking is one of the laborious job in fields. Weeds are the plants growing in a wrong place which compete with crop for water, light, nutrients and space, causing reduction in yield and effective use of machinery and can cause a disturbance in agriculture. Weeds can also host pests and diseases that can spread to cultivated crops.In olden days weed detection was done by employing some people, especially for that purpose. They will detect the weed by checking each and every place of the field. Then they will pluck them out manually using their hands.

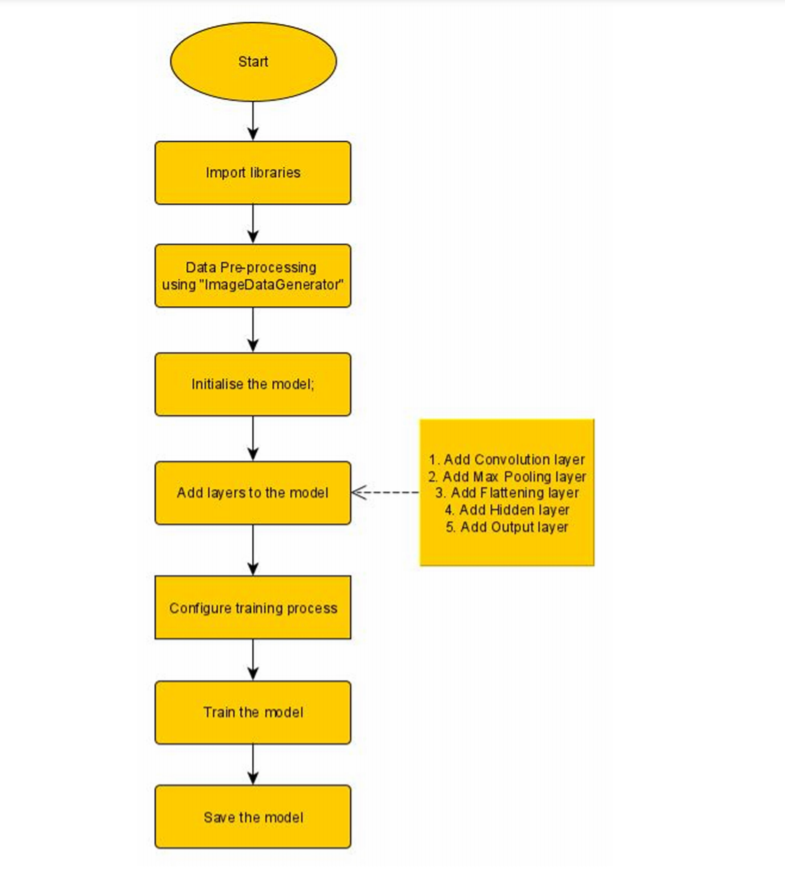
**2.2 Proposed Solution**

We are Proposing a solution in the device is integrated with camera and there will be a live video streaming in that it will detect the weed in the crop by using image processing. This system will distinguish the crop and weed. Our system will use a Convolution neural network algorithm to extract the features from image and train them by using neural network.

**3. Theoretical Analysis**

**3.1 Block Diagram**

The following is the block diagram of our proposed solution of using the Conolution Neural Networks to detect the weed whether present in the soil or not.



**3.2** **Hardware/Software Designing**

**Hardware Requirements:-**

Laptop/Desktop with webcam Software Requirements:-

• Python

• Keras

• Spyder

• Tensorflow

• Jupyter Notebook

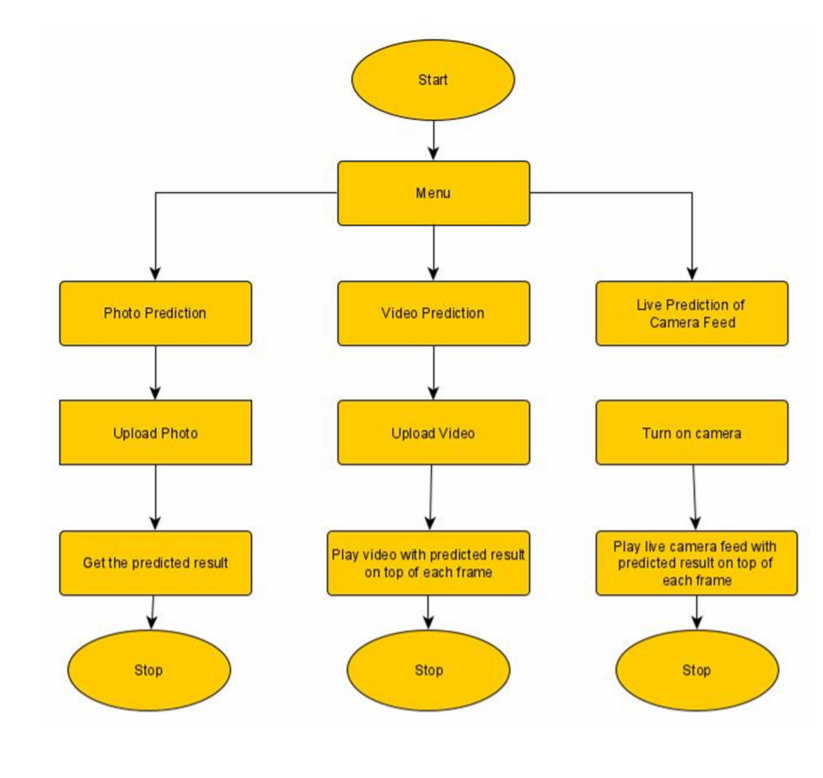
**4. Experimental Investigations**

• The crops in the field very much look alike, to achieve a practically good accuracy larger dataset is needed.

• The dataset need to be more distributed and unbiased to achieve to a accurate model.

• Edge Detection is quite harder with the crops dataset because of similarity of the crops with their background.

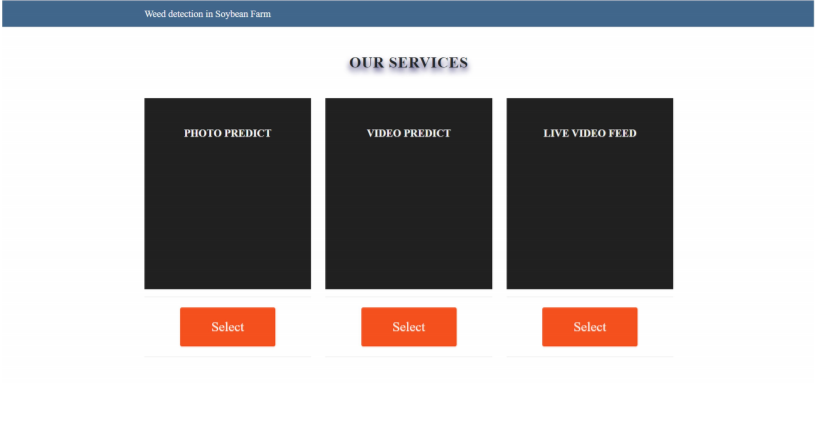
**5. Flowchart:**

****

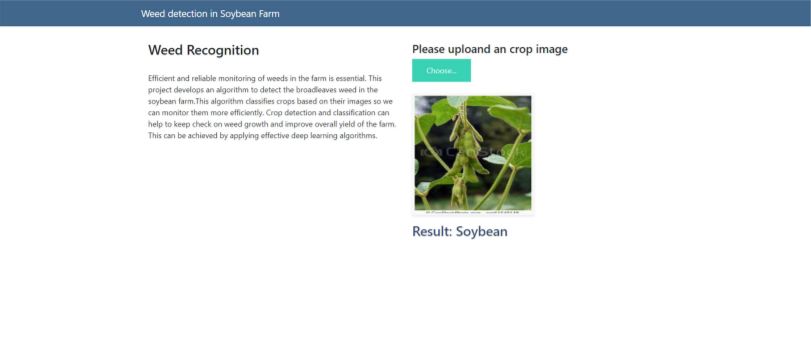
**6. Result**

The following images show the screenshots of our application of the **AI Based Weed Recognition System**

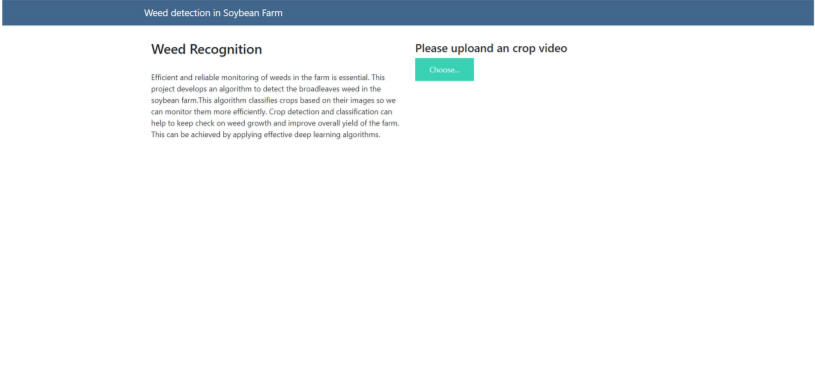
**Main Menu**



**Photo Predictor:**

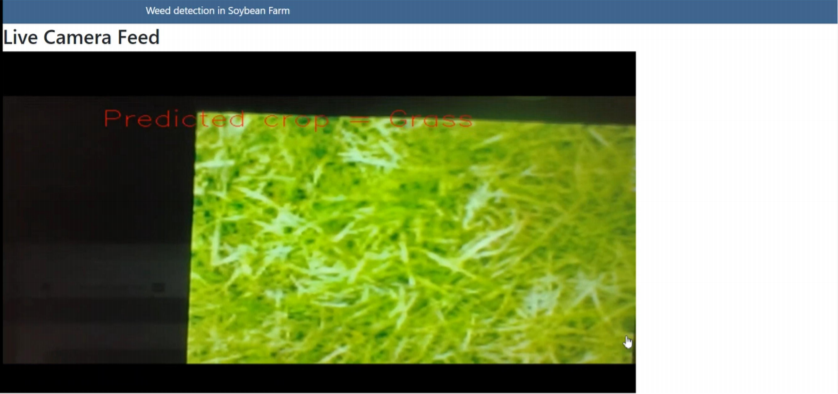


**Video Predictor:**





**Live Camera Feed Predictor**



|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Easy Model building with less formal statistical knowledge required. | Sharing an existing CNN model is difficult. |
| Capable of capturing non linearties between predictors and outcome | Prone to overfitting due to the complexity of model structure |

**7.Advantages and Disadvantages**

**8.Applications**

The AI enabled weed recognition system using Convolution neural networks is currently being practiced at many agricultural sectors. This application of CNN in weed recognition system helps many farmers as it is time saving, less work compared to the previous practices being done.

**9. Conclusion**

We would like to conclude that the developed model if trained with more data and better algorithms using more computation power can be deployed in a real world to help farmers and decrease the use of excessive herbicides , mitigating agricultural, environmental and health impact, and improving sustainability.

**10. Future Scope**

We can integrate this model to robot where the robot recognise the weed and pick automatically.

**11. References**

• <https://www.kaggle.com/fpeccia/weed-detection-in-soybean-crops>

• https://opencv-python tutroals.readthedocs.io/en/latest/py\_tutorials/py\_gui/py\_video\_display/py \_video\_display.html

**12.Appendix**

**Model Training**

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_ran ge=0.2,horizontal\_flip=True)

test\_datagen = ImageDataGenerator(rescale=1./255)

x\_train = train\_datagen.flow\_from\_directory(r"D:\Weed Project\dataset\trainset ",target\_size=(64,64),batch\_size=32,class\_mode='categorical') x\_test = train\_datagen.flow\_from\_directory(r"D:\Weed Project\dataset\testset",t arget\_size=(64,64),batch\_size=32,class\_mode='categorical')

Found 10837 images belonging to 4 classes.

Found 4499 images belonging to 4 classes.

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import Conv2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

model = Sequential() model.add(Conv2D(32,3,3,input\_shape=(64,64,3),activation='relu')) model.add(MaxPooling2D(pool\_size=(2,2))) model.add(Flatten()) model.add(Dense(output\_dim=128,activation='relu',init='random\_uniform')) model.add(Dense(output\_dim=4,activation='sigmoid',init='random\_uniform')) model.compile(optimizer='adam',loss='categorical\_crossentropy',metrics=['accuracy'])

print(x\_train.class\_indices) {'broadleaf': 0, 'grass': 1, 'soil': 2, 'soybean': 3} model.fit\_generator(x\_train,steps\_per\_epoch = 339,epochs=25,validation\_data= x\_test,validation\_steps = 141 )

Epoch 1/25

339/339 [==============================] - 55s 163ms/step - loss: 0. 5572 - accuracy: 0.7720 - val\_loss: 0.9070 - val\_accuracy: 0.8371

Epoch 2/25

339/339 [==============================] - 56s 164ms/step - loss: 0. 3995 - accuracy: 0.8477 - val\_loss: 0.3365 - val\_accuracy: 0.8860

Epoch 3/25

339/339 [==============================] - 61s 179ms/step - loss: 0. 3318 - accuracy: 0.8761 - val\_loss: 0.1398 - val\_accuracy: 0.8904

Epoch 4/25

339/339 [==============================] - 62s 182ms/step - loss: 0. 2766 - accuracy: 0.8955 - val\_loss: 0.1585 - val\_accuracy: 0.8969

Epoch 5/25

339/339 [==============================] - 60s 177ms/step - loss: 0. 2572 - accuracy: 0.9013 - val\_loss: 0.3267 - val\_accuracy: 0.8864

Epoch 6/25

339/339 [==============================] - 61s 180ms/step - loss: 0. 2416 - accuracy: 0.9051 - val\_loss: 0.5447 - val\_accuracy: 0.8128

Epoch 7/25

339/339 [==============================] - 64s 189ms/step - loss: 0. 2280 - accuracy: 0.9115 - val\_loss: 0.3420 - val\_accuracy: 0.8675

Epoch 8/25 339/339 [==============================] - 65s 192ms/step - loss: 0. 2288 - accuracy: 0.9143 - val\_loss: 0.1986 - val\_accuracy: 0.9235

Epoch 9/25

339/339 [==============================] - 63s 186ms/step - loss: 0. 2053 - accuracy: 0.9228 - val\_loss: 0.3976 - val\_accuracy: 0.9182

Epoch 10/25

339/339 [==============================] - 65s 190ms/step - loss: 0. 2061 - accuracy: 0.9195 - val\_loss: 0.3038 - val\_accuracy: 0.9164

Epoch 11/25

339/339 [==============================] - 62s 182ms/step - loss: 0. 1861 - accuracy: 0.9290 - val\_loss: 0.0540 - val\_accuracy: 0.9242

Epoch 12/25

339/339 [==============================] - 61s 179ms/step - loss: 0. 1742 - accuracy: 0.9318 - val\_loss: 0.0575 - val\_accuracy: 0.9233

Epoch 13/25

339/339 [==============================] - 61s 179ms/step - loss: 0. 1749 - accuracy: 0.9326 - val\_loss: 0.0283 - val\_accuracy: 0.9262

Epoch 14/25

339/339 [==============================] - 62s 183ms/step - loss: 0. 1616 - accuracy: 0.9383 - val\_loss: 0.2013 - val\_accuracy: 0.9222

Epoch 15/25

339/339 [==============================] - 62s 184ms/step - loss: 0. 1590 - accuracy: 0.9371 - val\_loss: 0.0789 - val\_accuracy: 0.9235

Epoch 16/25

339/339 [==============================] - 63s 187ms/step - loss: 0. 1585 - accuracy: 0.9404 - val\_loss: 0.1998 - val\_accuracy: 0.9224

Epoch 17/25

339/339 [==============================] - 63s 186ms/step - loss: 0. 1600 - accuracy: 0.9372 - val\_loss: 0.2660 - val\_accuracy: 0.9284

Epoch 18/25

339/339 [==============================] - 64s 189ms/step - loss: 0. 1485 - accuracy: 0.9450 - val\_loss: 0.3198 - val\_accuracy: 0.9231

Epoch 19/25

339/339 [==============================] - 62s 183ms/step - loss: 0. 1450 - accuracy: 0.9423 - val\_loss: 0.2404 - val\_accuracy: 0.9262

Epoch 20/25

339/339 [==============================] - 63s 186ms/step - loss: 0. 1457 - accuracy: 0.9428 - val\_loss: 0.0649 - val\_accuracy: 0.9178

Epoch 21/25

339/339 [==============================] - 62s 184ms/step - loss: 0. 1329 - accuracy: 0.9492 - val\_loss: 0.1291 - val\_accuracy: 0.9186

Epoch 22/25

339/339 [==============================] - 63s 186ms/step - loss: 0. 1270 - accuracy: 0.9516 - val\_loss: 0.2807 - val\_accuracy: 0.9255

Epoch 23/25

339/339 [==============================] - 62s 183ms/step - loss: 0. 1203 - accuracy: 0.9549 - val\_loss: 0.6384 - val\_accuracy: 0.9213

Epoch 24/25

339/339 [==============================] - 63s 185ms/step - loss: 0. 1373 - accuracy: 0.9480 - val\_loss: 0.2804 - val\_accuracy: 0.9255

Epoch 25/25

339/339 [==============================] - 63s 186ms/step - loss: 0. 1140 - accuracy: 0.9570 - val\_loss: 0.0202 - val\_accuracy: 0.9240

<keras.callbacks.callbacks.History at 0x2622d522988>

model.save("weed.h5")

**Flask app.py**

import numpy as np

import time

import os

import cv2

import tensorflow as tf

from keras.models import load\_model

from keras.preprocessing import image

from flask import Flask , request, render\_template, Response

from werkzeug.utils import secure\_filename

from gevent.pywsgi import WSGIServer

app = Flask(\_\_name\_\_)

model = load\_model("weed.h5")

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

def index():

if request.method == 'POST':

f = request.form["action"]

if(f=="photo"):

return render\_template('base.html')

elif(f=="video"):

return render\_template('video.html')

elif(f=="live"):

return render\_template('live.html')

@app.route('/')

def options():

return render\_template('index.html')

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

def upload():

if request.method == 'POST':

f = request.files['image']

print("current path")

basepath = os.path.dirname(\_\_file\_\_)

print("current path", basepath)

filepath = os.path.join(basepath,'uploads',f.filename)

print("upload folder is ", filepath)

f.save(filepath)

img = image.load\_img(filepath,target\_size = (64,64))

x = image.img\_to\_array(img)

x = np.expand\_dims(x,axis =0)

preds = model.predict\_classes(x) 4

print("prediction",preds)

index = ['Broadleaf','Grass','Soil','Soybean']

text = str(index[preds[0]])

return text

def uploadVideo(filepath):

video = cv2.VideoCapture(filepath)

fourcc = cv2.VideoWriter\_fourcc(\*'MP4V')

name = ["Broadleaf","Grass","Soil","Soybean"]

while(video.isOpened()):

success, frame = video.read()

if success==True:

cv2.imwrite("image.jpg",frame)

img = image.load\_img("image.jpg",target\_size = (64,64))

x = image.img\_to\_array(img)

x = np.expand\_dims(x,axis=0)

pred = model.predict\_classes(x)

p = pred[0]

print(pred)

cv2.putText(frame, "Predicted crop = "+ str(name[p]), (100,100), cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,0,255),2)

#print(frame.shape)

#cv2.imshow("image",frame)

frame = cv2.imencode('.jpg', frame)[1].tobytes()

yield (b'--frame\r\n'b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

if(cv2.waitKey(1) & 0xFF == ord('a')):

break

else:

break

@app.route('/video1',methods=['GET','POST'])

def video():

if request.method == 'POST':

f = request.files['video']

print("current path")

basepath = os.path.dirname(\_\_file\_\_)

print("current path", basepath)

filepath = os.path.join(basepath,'uploads',f.filename)

print("upload folder is ", filepath)

f.save(filepath)

return Response(uploadVideo(filepath),mimetype='multipart/x-mixedreplace; boundary=frame')

def gen():

"""Video streaming generator function."""

cap = cv2.VideoCapture(0)

# Read until video is completed

name = ["Broadleaf","Grass","Soil","Soybean"]

while(cap.isOpened()):

# Capture frame-by-frame

ret, frame = cap.read()

if ret == True:

cv2.imwrite("image.jpg",frame)

img = image.load\_img("image.jpg",target\_size = (64,64))

x = image.img\_to\_array(img)

x = np.expand\_dims(x,axis=0)

pred = model.predict\_classes(x)

p = pred[0]

print(pred)

cv2.putText(frame, "Predicted crop = "+ str(name[p]), (100,100), cv2.FONT\_HERSHEY\_SIMPLEX,1,(0,0,255),1)

frame = cv2.resize(frame, (0,0), fx=2, fy=1.5)

frame = cv2.imencode('.jpg', frame)[1].tobytes()

yield (b'--frame\r\n'b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

time.sleep(0.1)

else:

break

@app.route('/video\_feed')

def video\_feed():

"""Video streaming route. Put this in the src attribute of an img tag."""

return Response(gen(),

mimetype='multipart/x-mixed-replace; boundary=frame')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug = True, threaded = False)