

PROJECT DEVELOPMENT PHASE SPRINT 4

TEAM ID	PNT2022TMID46063
PROJECT NAME	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
DATE	13 NOVEMBER 2022

STEP 1: First open python code and run code, this capture the image in video and identify which animal or object are captured.

```
animal.py - C:\Users\murug\Desktop\opencv\animal.py (3.7.0)
File Edit Format Run Options Window Help

import cv2
import numpy as np

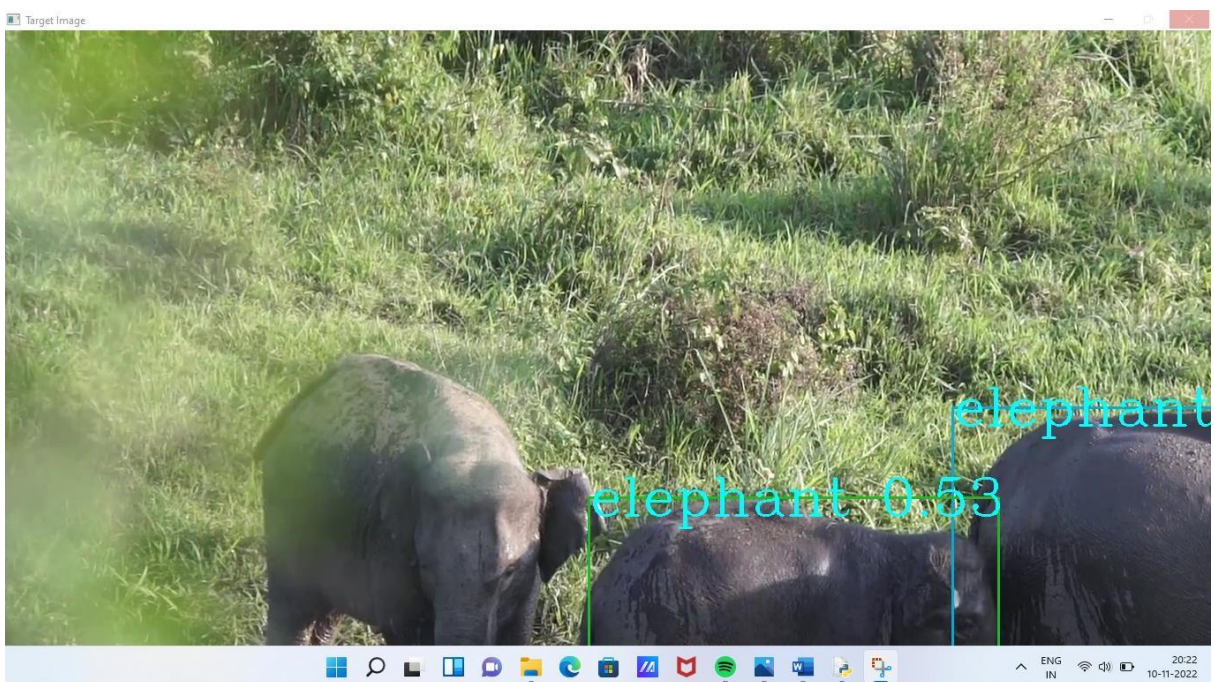
net=cv2.dnn.readNet('yolov3.weights','yolov3.cfg')
classes=[]
with open('classes.names','r') as f:
    classes=f.read().splitlines()
# img=cv2.imread('elephant.jpg')
cap=cv2.VideoCapture('video.mp4')
# cap=cv2.VideoCapture('person.jpg')
# cap=cv2.VideoCapture(0)
while True:
    _,img=cap.read()
    height,width,_=img.shape
    blob=cv2.dnn.blobFromImage(img,1/255,(416,416),(0,0,0),swapRB=True,crop=False) # (img,reduction the pixels size,size of the image,rgb colour)
    net.setInput(blob)
    output_layers_names=net.getUnconnectedOutLayersNames()
    layeroutput=net.forward(output_layers_names)
    boxes=[]
    confidences=[]
    class_ids=[]
    for output in layeroutput:
        for detection in output:
            scores=detection[5:]
            class_id=np.argmax(scores)
            confidence=scores[class_id]
            if confidence > 0.5:
                center_x=int(detection[0]*width)
                center_y=int(detection[1]*height)
                w=int(detection[2]*width)
                h=int(detection[3]*height)

                x=int(center_x - w/2)
                y=int(center_y - h/2)

                boxes.append([x,y,w,h])
                confidences.append(float(confidence))
                class_ids.append(class_id)

Ln: 14 Col: 13
```

STEP 2: It shows the detected animal or object name which is represented by square with the name of the animal or object.



PYTHON CODE:

```
import cv2
import numpy as np
```

```

net=cv2.dnn.readNetFromDisk('yolov3.weights',
yolov3.cfg')
classes=[] with
open('coco.names','r') as f:
classes=f.read().splitlines()

# img=cv2.imread('elephant.jpg')
cap=cv2.VideoCapture('video.mp4')
# cap=cv2.VideoCapture('person.jpg')
# cap=cv2.VideoCapture(0) while
True:    _,img=cap.read()

    height,width,_=img.shape

blob=cv2.dnn.blobFromImage(img,1/255,(416,416),(0,0,0),swapRB=True,crop
=False) #(img,reduction the pixels size,size of the image,rgb colour)
net.setInput(blob)
output_layers_names=net.getUnconnectedOutLayersNames()
layeroutput=net.forward(output_layers_names)    boxes=[]
confidences=[]    class_ids=[]    for output in layeroutput:        for
detection in output:            scores=detection[5:]
class_id=np.argmax(scores)            confidence=scores[class_id]

        if confidence > 0.5:
            center_x=int(detection[0]*width)
center_y =int(detection[1]*height)

```

```
w=int(detection[2]*width)
```

```
h=int(detection[3]*height)
```

```
x=int(center_x - w/2)
```

```
y=int(center_y - h/2)
```

```
boxes.append([x,y,w,h])
```

```
confidences.append((float(confidence)))
```

```
class_ids.append(class_id)
```

```
indexes=cv2.dnn.NMSBoxes(boxes,confidences,0.5,0.4)
```

```
font=cv2.FONT_HERSHEY_COMPLEX
```

```
colors=np.random.uniform(0,255,size=(len(boxes),3))
```

```
for i in indexes.flatten():
```

```
    x,y,w,h=boxes[i]    label=str(classes[class_ids[i]])
```

```
    confidence=str(round(confidences[i],2))    color=colors[i]
```

```
    cv2.rectangle(img,(x,y),(x+w ,y+h),color,2)    cv2.putText(img,label + "
```

```
    "+confidence,(x,y+20),font,2,(255,255,0),2)    cv2.imshow("Target
```

```
    Image',img)
```

```
    key=cv2.waitKey(1)
```

```
    if key ==ord('q):
```

```
        break cap.release() #
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```