IAES International Journal of Robotics and Automation (IJRA)
Vol. 99, No. 1, Month 2099, pp. 1~1x

Project Report:Tic Tac Toe

Yash Rahul Bellap
[1]Robotics And Autonomous Systems(Systems Engineering), Arizona State Univeristy, Az, USA

Article Info

Article history:

Received month dd, yyyy Revised month dd, yyyy Accepted month dd, yyyy

ABSTRACT

This Report talks about the procedures used to develop the Cobot to Play Tic Tac Toe with the User Using Yolo integration. This project was a learning experience which taught me many things i.e on how to create a Data set and how to intergrate YOLO in my code to move my robot.

.

Corresponding Author:
Yash Rahul Bellap
Robotics And Autonomous systems, Arizona State Univeristy Az, USA
Email: ybellap@asu.edu

1. Procedure:

- **Creating the Dataset:**

I used an image the base of the recognition area and took a random emoji(dog) to create random positions of the user input so I could train these data sets .

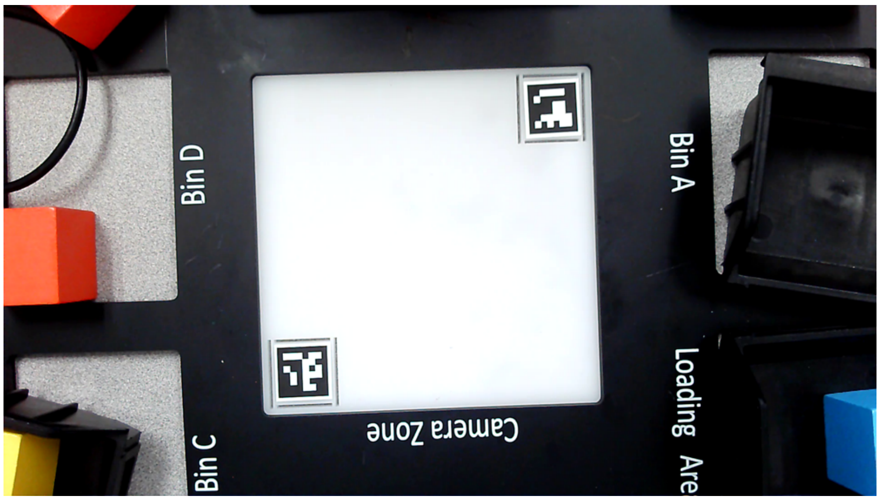


Image1: The Base of the Recognition Area

1.1. Task 1.1:

Step2: How to Create the Data Sets:

- After taking the dog emoji and creating the random positions on the base I Used Roboflow software to annotate the images and then generated the datasets which I then trained them with my code. Once these datasets were trained I then used random image positions while performing my experiment.

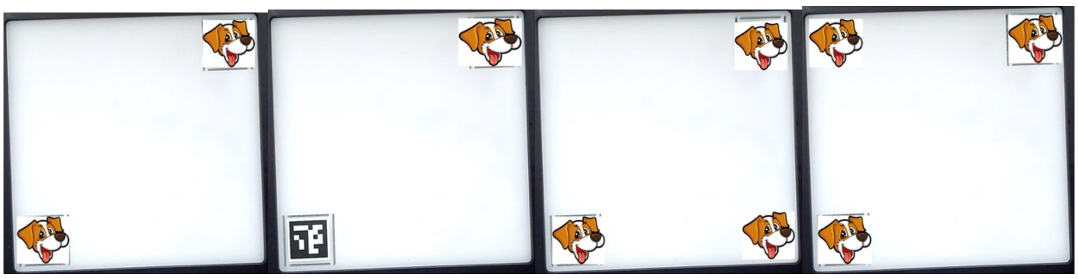


Image 2: Random positions of the user input before creating and training the data set.

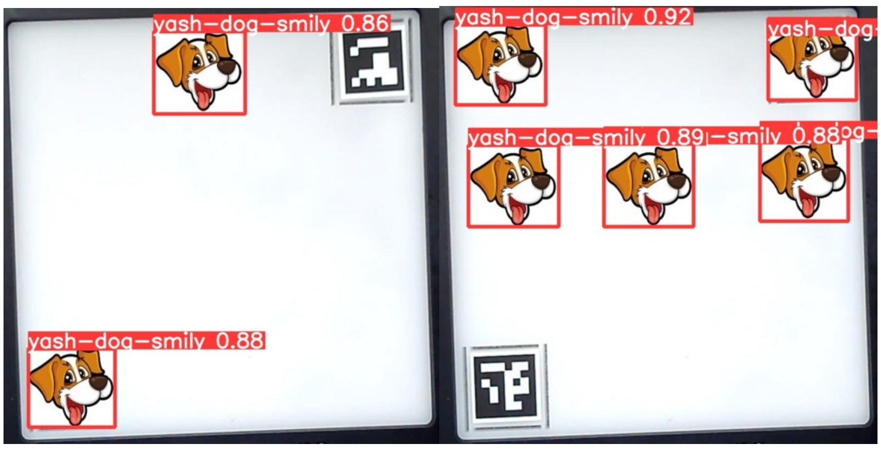
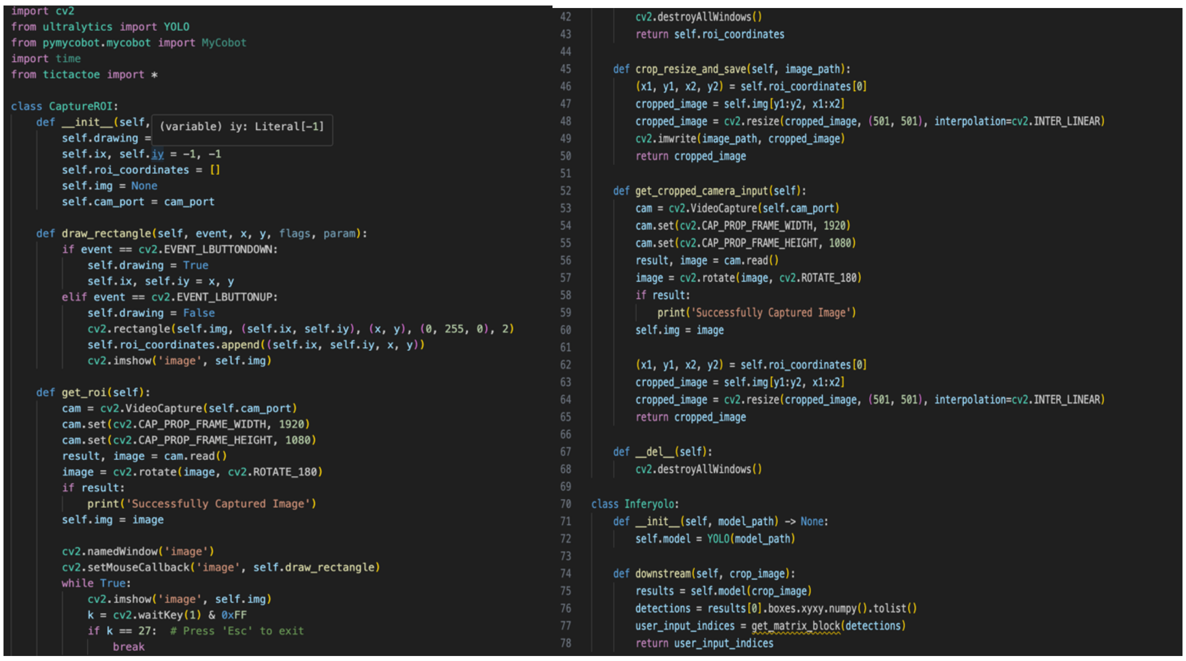
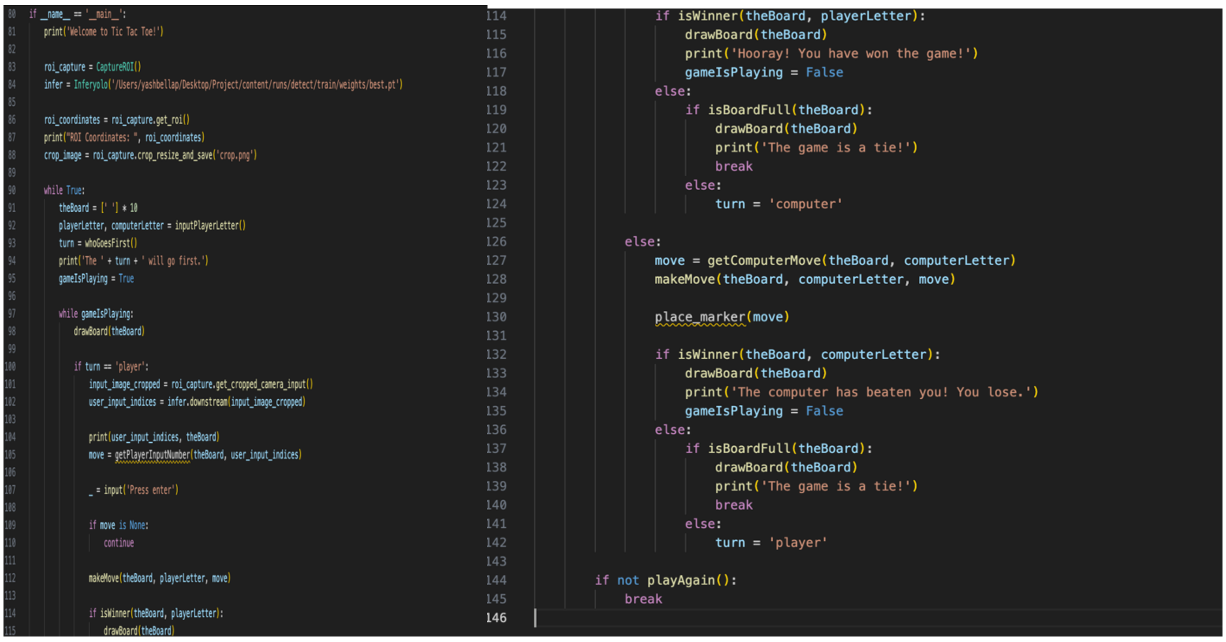


Image3 : After Training the Data set.



```python
import cv2
from ultralytics import YOLO
from pymycobot.mycobot import MyCobot
import time
from tictactoe import *

class CaptureROI:
    def __init__(self,
        self.drawing =
        self.ix, self.iy = -1, -1
        self.roi_coordinates = []
        self.img = None
        self.cam_port = cam_port

    def draw_rectangle(self, event, x, y, flags, param):
        if event == cv2.EVENT_LBUTTONDOWN:
            self.drawing = True
            self.ix, self.iy = x, y
        elif event == cv2.EVENT_LBUTTONUP:
            self.drawing = False
            cv2.rectangle(self.img, (self.ix, self.iy), (x, y), (0, 255, 0), 2)
            self.roi_coordinates.append((self.ix, self.iy, x, y))
            cv2.imshow('image', self.img)

    def get_roi(self):
        cam = cv2.VideoCapture(self.cam_port)
        cam.set(cv2.CAP_PROP_FRAME_WIDTH, 1920)
        cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
        result, image = cam.read()
        image = cv2.rotate(image, cv2.ROTATE_180)
        if result:
            print('Successfully Captured Image')
        self.img = image

        cv2.namedWindow('image')
        cv2.setMouseCallback('image', self.draw_rectangle)
        while True:
            cv2.imshow('image', self.img)
            k = cv2.waitKey(1) & 0xFF
            if k == 27:  # Press 'Esc' to exit
                break
```

```python
        cv2.destroyAllWindows()
        return self.roi_coordinates

    def crop_resize_and_save(self, image_path):
        (x1, y1, x2, y2) = self.roi_coordinates[0]
        cropped_image = self.img[y1:y2, x1:x2]
        cropped_image = cv2.resize(cropped_image, (501, 501), interpolation=cv2.INTER_LINEAR)
        cv2.imwrite(image_path, cropped_image)
        return cropped_image

    def get_cropped_camera_input(self):
        cam = cv2.VideoCapture(self.cam_port)
        cam.set(cv2.CAP_PROP_FRAME_WIDTH, 1920)
        cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
        result, image = cam.read()
        image = cv2.rotate(image, cv2.ROTATE_180)
        if result:
            print('Successfully Captured Image')
        self.img = image

        (x1, y1, x2, y2) = self.roi_coordinates[0]
        cropped_image = self.img[y1:y2, x1:x2]
        cropped_image = cv2.resize(cropped_image, (501, 501), interpolation=cv2.INTER_LINEAR)
        return cropped_image

    def __del__(self):
        cv2.destroyAllWindows()

class Inferyolo:
    def __init__(self, model_path) -> None:
        self.model = YOLO(model_path)

    def downstream(self, crop_image):
        results = self.model(crop_image)
        detections = results[0].boxes.xyxy.numpy().tolist()
        user_input_indices = get_matrix_block(detections)
        return user_input_indices
```



```python
if __name__ == '__main__':
    print('Welcome to Tic Tac Toe!')

    roi_capture = CaptureROI()
    infer = Inferyolo('/Users/yashbellap/Desktop/Project/content/runs/detect/train/weights/best.pt')

    roi_coordinates = roi_capture.get_roi()
    print("ROI Coordinates: ", roi_coordinates)
    crop_image = roi_capture.crop_resize_and_save('crop.png')

    while True:
        theBoard = [' '] * 10
        playerLetter, computerLetter = inputPlayerLetter()
        turn = whoGoesFirst()
        print('The ' + turn + ' will go first.')
        gameIsPlaying = True

        while gameIsPlaying:
            drawBoard(theBoard)

            if turn == 'player':
                input_image_cropped = roi_capture.get_cropped_camera_input()
                user_input_indices = infer.downstream(input_image_cropped)

                print(user_input_indices, theBoard)
                move = getPlayerInputNumber(theBoard, user_input_indices)

                _ = input('Press enter')

                if move is None:
                    continue

                makeMove(theBoard, playerLetter, move)

                if isWinner(theBoard, playerLetter):
                    drawBoard(theBoard)
                    print('Hooray! You have won the game!')
                    gameIsPlaying = False
                else:
                    if isBoardFull(theBoard):
                        drawBoard(theBoard)
                        print('The game is a tie!')
                        break
                    else:
                        turn = 'computer'

            else:
                move = getComputerMove(theBoard, computerLetter)
                makeMove(theBoard, computerLetter, move)

                place_marker(move)

                if isWinner(theBoard, computerLetter):
                    drawBoard(theBoard)
                    print('The computer has beaten you! You lose.')
                    gameIsPlaying = False
                else:
                    if isBoardFull(theBoard):
                        drawBoard(theBoard)
                        print('The game is a tie!')
                        break
                    else:
                        turn = 'player'

        if not playAgain():
            break
```

Image4: Image of my code.

NOTE: In the Video provided the dog images are the User Input X(x-tictactoe) and the green block is the Robots position playing as computer whose Turn displays O. As the Suction pump was not working I had to place the blocks of the robot manually.

Video Link:
https://drive.google.com/file/d/1mgP0LS1LuYXCxEzTCx0AUIDqhOUKcdcj/view?usp=drive_link