AN ENTITY IDENTIFICATION MECHANISM WITH TRACKING BY OPENCV AND YOLOV4 ALGORITHM

Anu Alexander
Department of Computer Applications
Amal Jyothi College of Engineering, Koovapally
Kanjirapally, Kerala
anualexander@mca.ajce.in

Navyamol K.T
Asst. Professor Department of Computer Applications
Amal Jyothi College of Engineering, Koovapally
Kanjirapally, Kerala
navyamolkt@amaljyothi.ac.in

Abstract: This paper consists of a methodology that is proposed to detect the object by the object detection algorithm from the Open Computer Vison (OpenCV) and YOLO Algorithm. The point in which this paper is to take the no of objects who are there in a frame. The packaging in which the video is stacked is done actually. It helps a lot with to count the no of objects it detects and count the objects it present.

Keywords: OpenCV, Object Detection Algorithm, YOLOV4 Algorithm

I. INTRODUCTION

It is the study of PCs and programming frameworks that perceive and comprehend the photos and casings. PC Vision is likewise made out of different angles. The most usually utilized piece of PC vision is the Object Detection because of the applications and its useful employments. Object discovery is likewise utilized for face location, vehicle recognition and following and so on OpenCV executes object location by the utilization of old-style calculations, the well-known PC vision library. The old-style calculations can't perform well to work under various conditions.

YOLO A calculation that distinguishes different articles in the casing (continuously). Object recognition in YOLO is a piece of the relapse issue. Just go for its calculation trains CNN to identify objects continuously. It says that the expectation in the whole picture is done in a solitary calculation run.

Consequences be damned calculation is significant on account of the accompanying reasons:
1. Speed: Improves the speed of identification
2. High precision: Provides exact outcomes with at most zero blunders.
3. Learning abilities: The calculation empowers to gain proficiency with the portrayals of items and apply the learned portrayal them in object recognition.

Simply take the plunge computation works using the going with three techniques:
1. Residual squares: The picture partitions to matrices. Every matrix contains the element of S x S.
2. Bounding box relapse: A blueprint that features an item in a picture. The picture comprises of the accompanying credits: Width, Height, Class, Bounding box focus.
1. Intersection over association (IOU): Provide a result box that encompasses the identified articles flawlessly. Every framework cell is for expecting the skipping boxes and their relating sureness scores. On the off chance that this strategy is equivalent to 1 the anticipated bouncing box is as old as genuine box.

The course of Object following comprises of:
1. Taking an info set of bouncing box arranges.
2. Appointing an extraordinary ID for every one of its underlying directions.
3. Following every one of the articles move around outlines in a casing.
4. Keeping up with the remarkable IDs

II. LITERATURE REVIEW

Bhumika Gupta (2017) proposed object acknowledgment is Related known PC advances Focused on PC vision and picture handling Discovery of an object of a specific class or its example (People, blossoms, creatures in computerized pictures and recordings). Object recognition has several uses. Well researched on face recognition, characters, etc. Detection and vehicle computer. Object recognition Used for various purposes such as search and search monitoring. In this review, some essential ideas Object acknowledgment when utilizing OpenCV library from Python 2.7 works on the productivity and precision of articles Detection was introduced.

Kartik Umesh Sharma (2017) proposed an article disclosure structure finds objects of this current reality present either in a high level picture or a video, where the item can have a place with any class of articles in particular people, vehicles,and so on In request to recognize an item in a picture or a video the framework requirements to have a couple of parts to finish the undertaking of distinguishing an item, they are amodel data set, a include identifier, a hypothesizer and a hypothesizer verifier. This paper presents an audit of the different procedures that are utilized to distinguish an item, limit an object, sort an item, extricate highlights, appearancedata, and some more, in pictures and recordings. The remarksare drawn dependent on the concentrated on writing and central questions are additionally recognized applicable to thearticle identification.
Geetha Priya. S (2019) proposed the Objective is to recognize of items utilizing You Only Look Once (YOLO) approach. This technique enjoys a few benefits as contrasted with other article recognition calculations. In other calculations like Convolutional Neural Network, Fast Convolutional Neural Network the calculation won't look at the picture totally however in YOLO the calculation looks the picture totally by

DOI: 10.5281/zenodo.6375233
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

foreseeing the jumping boxes utilizing convolutional network and the class probabilities for these cases and recognizes the picture quicker when contrasted with different calculations.

III. IMPLEMENTATION

The basic idea is to address an item's visual appearance as an incorrectly coordinated blend of various area setting regions keyed by specific key features, or pieces. Currently, the part extraction cycle will find a piece of these specific keys under various conditions. regardless, in regular only one out of every odd one of them. To be sure, even with the close by legitimate affirmation, such keys probably could be unsurprising with different overall hypotheses. However, whenever the as a whole fact is accumulated, the portion which can be found by known to be involved separation process is frequently sufficient to recognize objects in an image. This is one of the rule issues of thing affirmation, which is that, in any case, the circumstances are rather imaginary, it has as yet exhibited hard to constantly part whole articles on a bottom-up premise. In a current system, close by features rely upon normally removed cut-off parts are used to address distinctive 2•D points of view (portions) of inflexible 3•D things, but the fundamental idea can be applied to various components and various depictions.

IV. IMPLEMENTATION METHODS

A. Consequences be cursed Weights in certified article acknowledgment:

You simply look once (YOLO) is a thing acknowledgment system used for steady picture taking care of.

YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Figure 1.1 Types of YOLO weights

1. Advantages of YOLO:

a) Speed (45 housings each second), speedier than continuous
b) Network grasps summarized object depiction
c) It is a quicker form with a more modest design of 155 casings for each sec, yet is less precise.

2. Limitations of YOLO:

Just go for it forces solid spatial imperatives on the jumping box expectations since every one of the matrix cells just predicts two boxes and can have just one class. This spatial requirement then, at that point, restricts the quantity of adjacent articles that our model can anticipate. The model battles with the little objects that show up in gatherings. Since the model figures out how to anticipate bouncing boxes from information, it anyway battles to sum up objects in new or surprising viewpoint arrangements

2. Webcam/Camera Installation:

It's done by utilizing OpenCV and We want to call the capacity cv2.VideoCapture(), and peruse the approaching edges. Seeing as the.read() technique is an obstructing activity; the basic string of our Python script is totally obstructed until the edge is read from the camera gadget and returned to our content. This is an issue but it is essential for our framework to run indefinitely. We can improve the FPS (frames per second) merely by launching a unique string that sits idle but pulls the camera for acquired signals while our primary string handles handling over the current edge.

3. Evaluation:

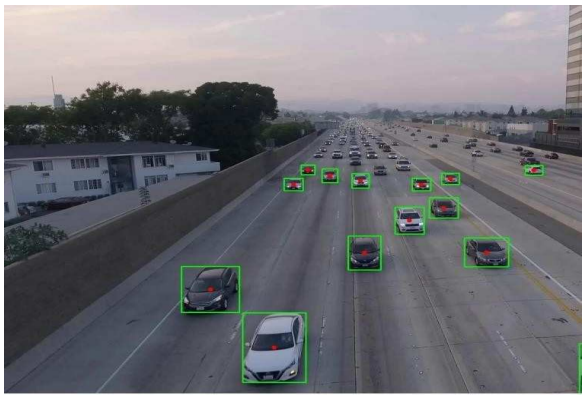
Counting the extra visual hints that may diminish the pace of bogus negatives to an even lower number. A little tendency of a walkway presents little mistakes in a distance gauge. This isn't actually an issue as the mistakes are little for hindrances that are near the framework. Our framework anyway misjudges the distance to overhanging impediments, which disregards the third supposition. This distance can assess blunder effectively and lead to crashes, in any event, when the pixels are accurately named hindrances.

A. General Algorithm of the system:

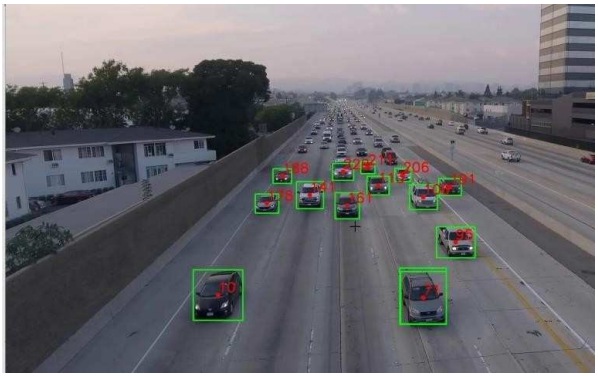
Steps	Description
Step I	Consider an image and we need to create a grid that will give us the features of an object.
Step II	In this step, we make using of OpenCV which will read the input image and data points and specify the file path to an image in a Numpy array.
Step III	Detecting an image in a grid view after the process of reading image by OpenCV and Numpy and converting grid to rectangular boxes.
Step IV	The final step consists of displaying the image with the rectangular box along with the caption on the window. This is done using YOLO and COCO dataset.

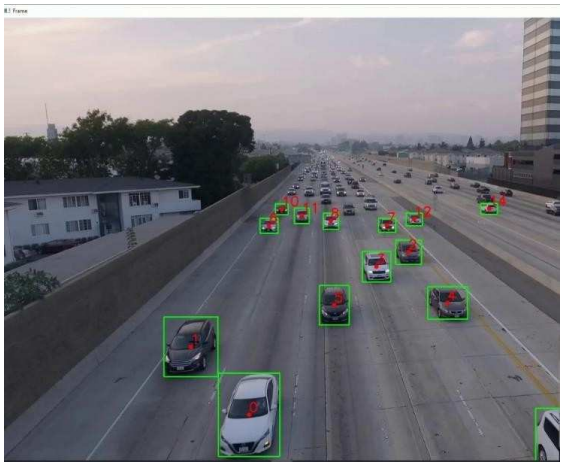
B. Algorithm for the making of frames:

```
from object detection import Object Detection
# Initialize Object Detection
od = ObjectDetection ()
while True:
  # Detect objects on frame
  (class_ids, scores, boxes) = od.detect(frame)
  for box in boxes:
    (x, y, w, h) = box
    cx = int ((x + x + w) / 2)
    cy = int((y + y + h) / 2)
    center_points_cur_frame.append((cx, cy))
```



DOI: 10.5281/zenodo.6375233
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

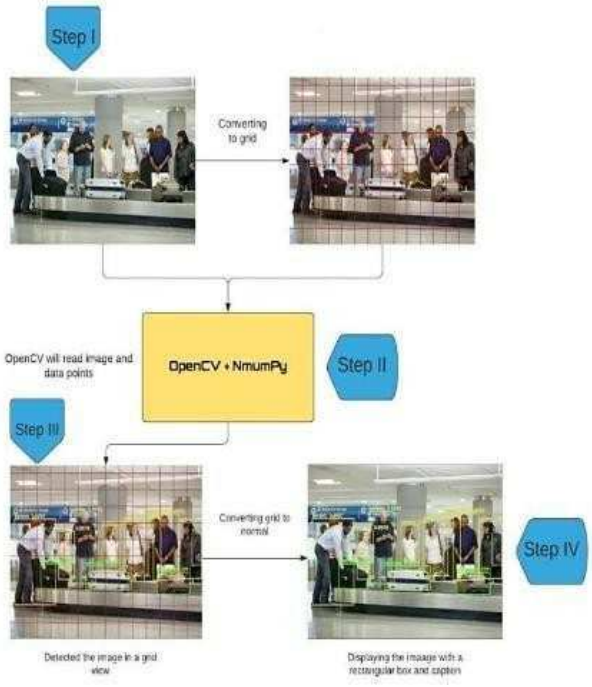




C. *Algorithm for Object Track*

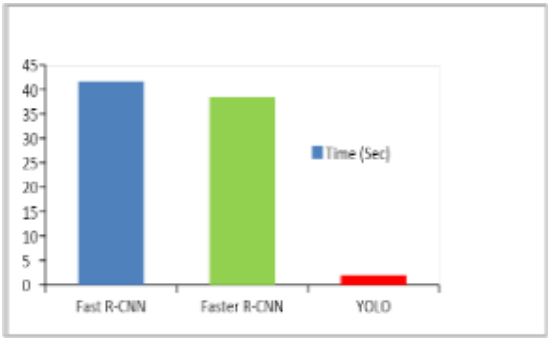
1. Find the point and assign the ID

```
# Initialize count
count = 0
center_points_prev_frame = []
tracking_objects = {}
track_id = 0
    # Only at the beginning we compare previous and current
frame
    if count <= 2:
        for pt in center_points_cur_frame:
            for pt2 in center_points_prev_frame:
                distance = math.hypot(pt2[0] - pt[0], pt2[1] - pt[1])
                if distance < 20:
                    tracking_objects[track_id] = pt
                    track_id += 1
```



2. Add new ID to the objects

```
else:
        tracking_objects_copy = tracking_objects.copy()
        center_points_cur_frame_copy=
center_points_cur_frame.copy()
        for object_id, pt2 in tracking_objects_copy.items():
            object_exists = False
            for pt in center_points_cur_frame_copy:
                distance = math.hypot(pt2[0] - pt[0], pt2[1] - pt[1])
                # Update IDs position
                if distance < 20:
                    tracking_objects[object_id] = pt
                    object_exists = True
                    if pt in center_points_cur_frame:
                        center_points_cur_frame.remove(pt)
                    continue
# Remove IDs lost
            if not object_exists:
                tracking_objects.pop(object_id)
        # Add new IDs found
        for pt in center_points_cur_frame:
            tracking_objects[track_id] = pt
            track_id += 1
```



V. CONCLUSION

In light of experimental outcomes, the item can be recognized all the more precisely and separately related to the specific area of an article in the picture alongside the x and y-hub. This paper gave exploratory outcomes on various techniques for object discovery and distinguishing proof and thought about every

DOI: 10.5281/zenodo.6375233
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

strategy for their efficiencies, additionally performed effective item identification while not thinking twice about the performance. The time take to detect the algorithm is also described below with an image.

VI. FUTURE WORK

We can implant the work in TensorFlow has much better and simpler help for saving and stacking models across very surprising conditions and in any event, programming dialects. Assuming stacking and saving models is vital, then, at that point, TensorFlow proves to be useful.

VII. REFERENCES

1 Moon, J. H. Kim and W. S. Lee, "Progressed Lane Recognizing Algorithm for Unmanned Vehicle," Proceeding of the ICCAS2003, pp. 1130-1133, 2003.

2 H. Park, Y. J. Child and J. H. Kim, "Plan of Advanced Tele-worked Control System for Unmanned Vehicle," Continuing of the ICCAS2005, pp. 915-919, 2005.

3 Cosmin D. Pantiliner, Silvius Bota, Istvan Haller, Sergiu Tedeschi, "Ongoing Obstacle Detection Using Dense Sound System Vision and Dense Optical Flow", 2010, ISBN 978-1-4244-8230-6/101\$26.00.

4 Ryuzo Okada, Yasuhiro Taniguchi, Kenji Furukawa, Kazunori Onoguchi, "Impediment Detection Using Projective Invariant and Vanishing Lines", IEEE Worldwide Conference in Computer Vision (ICCV, vol. 2, 2000

DOI: 10.5281/zenodo.6375233
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

DOI: 10.5281/zenodo.6375233
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam

DOI: 10.5281/zenodo.6375233
ISBN: 978-93-5607-317-3 @2022 MCA, Amal Jyothi College of Engineering Kanjirappally, Kottayam