IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 6.252
Volume 2, Issue 9, June 2022

Visual Perception using OpenCV

Arun J[1], K G Prashanth[2], Nivedita MB[3], Nithin B[4], Dr. Shiv Prasad KM[5]
Students, B.E 4th Year, Department of Computer Science and Engineering[1,2,3,4]
Assisstant Professor, Department of Computer Science and Engineering[5]
Rao Bahadur Y Mahabaleswarappa Engineering College Bellary, Karnataka, India
Affiliated to Visvesvaraya Technological University, Ballari, Karnataka, India

Abstract: *Visual Perception is related with detecting objects. Detection of the object is done by using YOLO network. There are multiple object detection algorithms like Fast- Convolutional Neural Network (Fast-CNN), Faster-Convolutional Neural Network(Faster- CNN), Regional- Convolutional Neural Network (R-CNN) that does not process the whole image at a time, but when compared to these, YOLO looks at the image completely and then passes it to the single network, which then predicts the bounding boxes using convolutional network and class probabilities for these bounding box and detects the image faster and accurately.*

Keywords: Fast-CNN, R-CNN

I. INTRODUCTION

Humans can easily recognise and detect the objects present in an image. The visual system of humans is fast and more precise when compared to trained systems and can perform the intricate tasks of identifying numerous objects in an image and detect the obstacles with a little conscious thought. The systems can be trained to identify and label numerous objects in an image with high speed and precision, with the availability of large amount of data, faster GPU's and improved algorithms.

Article Discern is concerned with detecting numerous objects in a real time image, a video sequence or a static image.

Article Discern and tracking algorithms can be divided into traditional methods and deep learning methods. These algorithms can latter be roughly classified into region proposal object detection algorithms like R- CNN, SPP net, Fast-CNN and Faster-CNN which generates the region proposals and classifies them afterwards into region proposals. The other algorithms are based on regression object detection algorithms namely SSD and YOLO that generates and classifies the region proposals at the same interval.

In spite of that, these algorithms are not tested with degraded pictures simply saying they are tested with the academic datasets like ImageNets, COCO and VOC, although they are not tested with datasets captured randomly. The main issue with the pictures captured in real time include:

1. Image captured can be blurred because of the camera instability.
2. Image obtained may not be clear because of the object obstruction

Bad weather condition, overexposure or low resolution can lead to the poor resolution of image.

Figure 1.2: Image problems under exposure, blur, noise

In traditional approaches, a sliding window was implemented for identifying the items in an image at different locations and scales. This operation was quite expensive due to which the aspect ratio of objects was assumed to be constant.

The object detection algorithms based on deep learning approach namely R-CNN, Fast-CNN and Faster-CNN used a selective search technique to discard the redundant bounding boxes and passed to the algorithm for testing.

Later YOLO was introduced to handle the detection of an object in a completely discrete way. This forwards the entire image only once through the network.

This network divides the image into numerous regions and allots the bounding boxes for these regions along with the probabilities.

YOLOv3 when relatively compared to other algorithms is extremely fast. In this, the non maximum suppression(NMS) is used to lessen the number of redundant bounding boxes for the same object. The bounding boxes with a value less than the threshold confidence level are eliminated and the one with higher level, outputs recognised objects along with the bounding boxes.

II. RELATED WORK

You Only Look Once: Unified, Real-Time Object Detection, by Joseph Redmon. Their prior work is on detecting objects using a regression algorithm. To get high accuracy and good predictions they have proposed YOLO algorithm in this paper [1].

Understanding of Object Detection Based on CNN Family and YOLO, by Juan Du. In this paper, they generally explained about the object Revised Manuscript Received on February 14, 2019 detection families like CNN, R-CNN and compared their efficiency and introduced YOLO algorithm to increase the efficiency [2].

Learning to Localize Objects with Structured Output Regression, by Matthew B. Blaschko. This paper is about Object Localization. In this, they used the Bounding box method for localization of the objects to overcome the drawbacks of the sliding window method [3].

III. IMPLEMENTATION

3.1 Object Detection

The algorithm implemented in this paper detects the objects taken from a real-time image, a video sequence and a static image. Detection of objects involves passing through several steps as shown below.

For this purpose we require 3 main files:

- yolov3.cfg–which is a configuration file
- yolov3.weights–pre-trained weights
- coco.names–names of the 80 classes used in coco dataset

Firstly we load an image and reduce the height and width of an image to the scale of 40% to 30%. Then we save all the values of these in height, width channels variables for the original image.

This will be the original image from which we want to detect as many objects as possible. But this image cannot be directly passed through the algorithm.

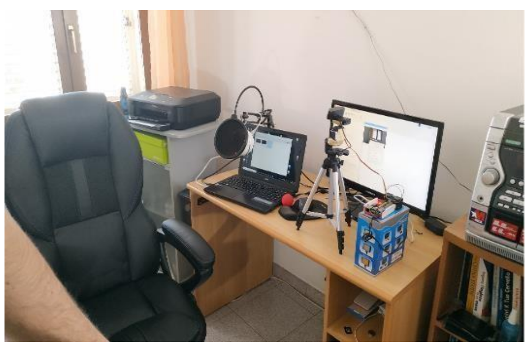

Figure 3.1: Original image

So, we have to convert this image into a certain format called as Blob, which is basically concerned with extracting features from the image. Objects in blob are detected using **cv2.dnn.blobFromImage** by passing few variables like: img as filename, scale factor of 0.00392 and the image size in blob should be (416,416) with o mean subtraction (0,0,0) and the True flag means we will be inverting blue to red since OpenCV uses BGR and the image we have is in RGB.

DOI: 10.48175/IJARSCT-5355

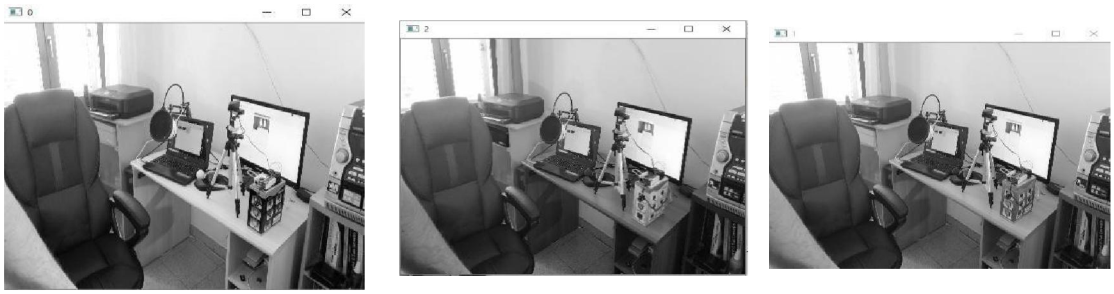

Figure 3.2: Blob images

After this, we now pass the blob image to the network using **net.setInput(blob)** which will then be forwarded to the output layers.

Here all the objects from the image and the outs contain all the information needed to extract the position of the object like top, bottom, left, right.

Further, we will be evaluating the outs so here we will be trying to predict confidence. For this, firstly we will loop through each outs and get scores for each out in outs.

Later we get the class_id and assign the highest confidence score by passing the class_id.

After this a threshold value for confidence score is set as 0.5, which means that the objects are detected from an image when they surpass this threshold value.

The blob output image obtained is then passed as an input through the network for processing and a forward pass is run to get the list of predicted bounding boxes as an output. We need to find out the last layers of the network to run through the whole network which is done by using the **getUnconnectedOutLayers()** function. Then we obtain the output from the output layers of the network by running forward pass.

There might be a cases for detecting the same object many number of times as shown below. To eliminate this we will be using non-Maximum Suppression(NMS) which is controlled by **nmsThreshold** parameter.

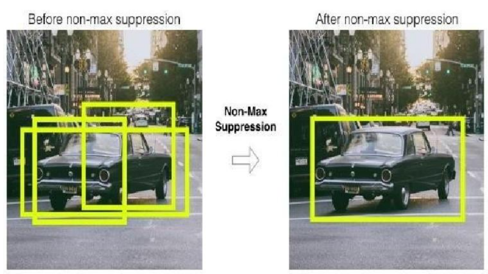

Figure 3.3: Non-Maximum Suppression

Finally, we draw the boxes filtered through the non-maximum suppression, on the input frame with their assigned class labels.

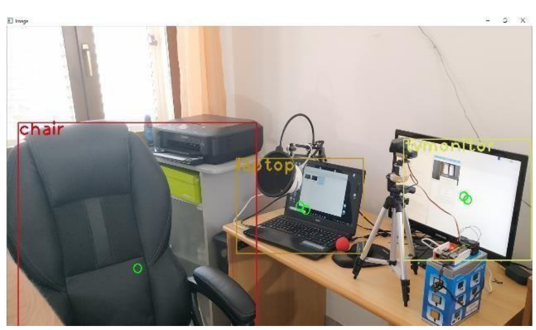

Figure 3.4: Output Image

3.2 Object Tracking

Object tracking is a field within computer vision that involves tracking objects as they move across several video frames. Objects may often be people, but they can also be animals, vehicles or other objects of interest such as ball in the game of cricket, tennis or soccer. It has many practical applications including surveillance, traffic flow analysis, etc.. Object tracking starts with detection of the object and assigning the bounding box for each object and tracking the path of the object. If there is a movement in video sequence then the bounding boxes are assigned, else bounding boxes are not assigned for the static objects in the video.

3.3 Object Search

The searching has been implemented to know whether the provided name of an object is present in a static image or in a real-time image. The name of the object to be searched is provided, the trained system scans the objects in image, extracts the features and maps it with the trained model features. If the object is detected, then the window is closed automatically and prints the status of the object as available or not.

IV. EXPERIMENTAL RESULTS

Fig 4.1: Home page

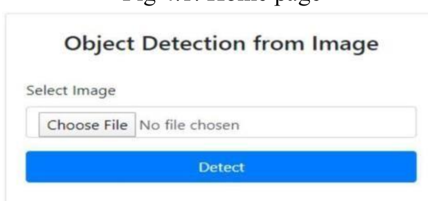

Fig 4.2: Detection from image

Figure 4.3: Results obtained

Figure 4.4: Detected Object Name

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 6.252
Volume 2, Issue 9, June 2022

Figure 4.5: Detection of object from Live

Figure 4.6: Results obtained

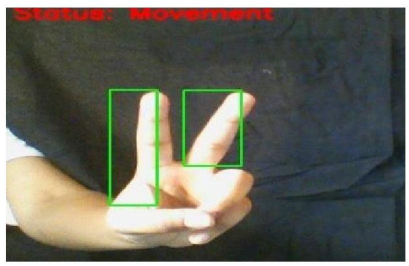

Figure 4.7: Object tracking Live

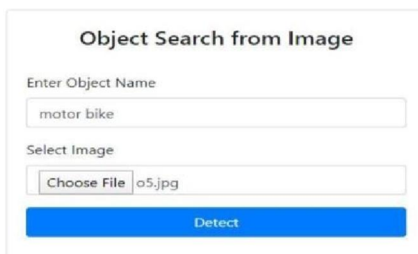

Figure 4.8: Object search from Image

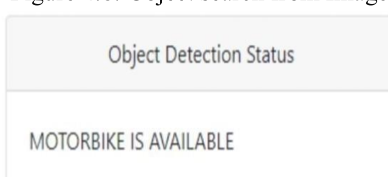

Figure 4.9: Results Obtained

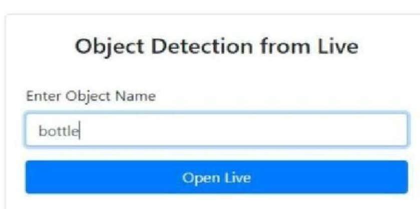

Figure 4.10: Object detection from Live

ISSN (Online) 2581-9429

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 9, June 2022

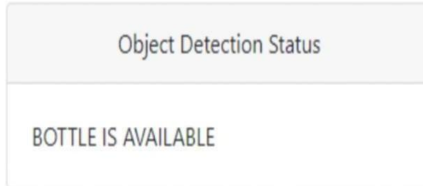

Figure 4.11: Results obtained

V. CONCLUSION

In this paper, we use yolo network that processes the image completely when passed through a single neural network in real-time scenarios .This model shows the excellent detection , tracking and searching results on the objects trained. Furthermore it can be utilized in specific scenarios to detect, track and respond to particular targeted objects in an video surveillance.

REFERENCES

[1]. Joseph Redmon, Santosh Divvala, Ross Girshick, “You Only Look Once: Unified, Real-Time Object Detection”,The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[2]. YOLO Juan Du1,”Understanding of Object Detection Based on CNN Family”, New Research, and Development Center of Hisense, Qingdao 266071, China.

[3]. Matthew B. Blaschko Christoph H. Lampert, “Learning to Localize Objects with Structured Output Regression”, Published in Computer Vision – ECCV 2008 pp 2-15.

[4]. Wei Liu, Dragomir Anguelov, Dumitru Erhan, “SSD: Single Shot MultiBox Detector”, Published in Computer Vision – ECCV 2016 pp 21- 37.

[5]. Chandan G,Ayush Jain, Harsh Jain,Mohana” Real Time Object detection using DL and OpenCV”.

[6]. Chengi Liu, Yufan Tao, Jiawei Liang, Kai Li, Yihang Chen, “Object Detection Based On YOLO Network”, ITOEC 2018.

[7]. Geethapriya S N, Duraimurugan S P, Chokkalingam, “Real-Time Object Detection With YOLO”, International Journal of Engineering and Advanced Technology(IJEAT), ISSN:2249-8958, Volume-8, Issue-3S, Feb 2019

[8]. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature

[9]. Ross Girshick, Fast R-CNN. In international Conference on Computer Vision(ICCV), 2015.

[10]. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster RCNN:Towards real-time object detection with region proposal networks”.In Advances in Neural Information Processing Systems(NIPS), 2015.

[11]. Karen Simonyan and Andrew Zisserman.”Very deep convolutional networks for large-scale image recognition”.arXiv preprint arXiv:1409.1556,2014

[12]. Lichao Huang, Yi Yang, Yafeng Deng, Yinan Yu DenseBox, “Unifying Landmark Localization with End to End Object Detection”, Published in Computer Vision and Pattern Recognition (cs.CV).

[13]. Dumitru Erhan, Christian Szegedy, Alexander Toshev, “Scalable Object Detection using Deep Neural Networks”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2147-2154.

[14]. Joseph Redmon ,Ali Farhadi, “YOLO9000: Better, Faster, Stronger”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263-7271.

[15]. Jifeng Dai, Yi Li, Kaiming He, Jian Sun, “R-FCN: Object Detection via Region-based Fully Convolutional Networks”, published in: Advances in Neural Information Processing Systems 29 (NIPS 2016).