International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

LIVE OBJECT DETECTION USING OPENCV

A. Sri Varsha[1], B. Srinija[2], M. Mounika[3]
T. Monika Singh[4]

Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Telangana, India

Abstract.

Object detection is one of the most basic and central task in computer vision. Its task is to find all the interested objects in the image, and determine the category and location of the objects. Object detection is widely used and has strong practical value and research prospects. Applications include face detection, pedestrian detection and vehicle detection. In recent years, with the development of convolutional neural network, significant breakthroughs have been made in object detection. The latest research in this field has been making tremendous development in many areas. Object detection and tracking have a variety of uses, this project presents a general trainable framework for object detection in images and videos including live video. The detection technique we are using is based on YOLO. In this project, we also discuss current and prospective applications of object detection in several fields. The results presented here suggest that this architecture can be further developed and used in face detection, face recognition, anomaly detection, crowd counting, security surveillance, etc. The Objective is to detect of objects using You Only Look Once (YOLO) approach. This method has several advantages as compared to other object detection algorithms. In other algorithms like Convolutional Neural Network, Fast Convolutional Neural Network the algorithm will not look at the image completely but in YOLO the algorithm looks the image completely by predicting the bounding boxes using convolutional network and the class probabilities for these boxes and detects the image faster as compared to other algorithms.

Keywords: Object detection, YOLO, Image processing, Computer vision, Machine learning, Training models.

1. Introduction

1.1 About Project

"Object Detection" is a dynamic application. The main disadvantage in the existing system R-CNN i.e., Region Based Convolutional Neural Networks where we need to classify huge numbers of regions for the detection. Existing System are outdated, for object detection and resource consuming. It can't be implemented real time as it takes around 47 seconds for each test proposal. Requirements are far greater than solutions available to store the feature map of the region proposals. So it takes a lot of time to train. To avoid all these limitations and allows to implement it in real world, the system needs to be replaced with a better system. Proposed system work with 2000 regions only which are generated using selective search algorithm. The aim of the proposed system is to eliminate the time consumption. All the previous object detection algorithms have used regions to localize the object within the image. The network does not look at the complete image. Instead, parts of the image which has high probabilities of containing the object. YOLO or You Only Look Once is an object detection algorithm much is different from the region – based algorithms. The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. Yolo Object Detection and Open CV allows the user to determine the functionality of the application.

1.2 Objectives of the Project

The main purpose of object detection is to identify and locate one or more effective targets from still image or video data. Object Detection in a moving video stream is playing a prominent role in every branch of science and research. Image classification involves assigning a class label to an image, whereas object localization involves drawing a bounding box around one or more objects in an image.

- Develop a computational model to identify the moving objects
- Detect the moving objects in various scenarios
- Develop a comparative result of efficiency for better object detection
- Develop an application for the smart surveillance system using object detection

1.3 Scope of the Project

Object Detection finds its scope in fields like

Vision-Based Control Systems : allows us to identify and locate objects in an image or video.

Human Computer Interface : is the task of identifying the physical movement of an object in a given region.

Medical Imaging : is one of the quickest growing bottlenecks in the medical world.

Augmented Reality : to identify the form and shape of different objects and their position in space caught by the device's camera.

Robotics : Object recognition and tracking reduces human efforts and provides efficiency.

2. Literature Survey

2.1 Existing System

Convolutional Neural Networks (CNNs) is neural network model being used for image classification problem. A CNN makes predictions by looking at an image and then checking to see if certain components are present in that image or not. If they are, then it classifies that image accordingly. CNN's have been extensively used to classify images. But to detect an object in an image and to draw bounding boxes around them is a tough problem to solve. Existing System are outdated, for object detection and resource consuming.

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The Convolution Neural Network has the following drawbacks:

- A Convolutional neural network is significantly slower due to an operation such as maxpool.
- If the CNN has several layers then the training process takes a lot of time if the computer doesn't consist of a good GPU.
- A ConvNet requires a large Dataset to process and train the neural network.
- Lack of ability to be spatially invariant to the input data.
- CNN do not encode the position and orientation of object.

2.2 Proposed System

To tackle these problems of the object detection, machine learning and deep neural network methods are more effective in correcting object detection. A modified new network is proposed based on the YOLOv4 network model. The proposed model effectively extracts features from images, performing much better in object detection. This system overcomes the issue of CNN model by detecting the objects, even when they are overlapping.

YOLO is an algorithm that uses neural networks to supply real-time object detection. This algorithm is popular due to its speed and accuracy. it's been utilized in various applications to detect traffic signals, people, parking meters, and animals. YOLO uses a special approach. YOLO may be a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm implements one neural network to the entire image, then breaks the image into sections and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the anticipated probabilities. In YOLO, a CNN predicts multiple bounding boxes at one given time and probabilities for those boxes. It trains on real images and directly optimizes performance.

Compared to other Region-Based Convolutional Neural Networks which perform detection on various regions and thus find yourself performing prediction multiple times for various regions in a picture or a video, Yolo's architecture is alike to FCNN hence YOLO passes the image once through the FCNN and output is the prediction. This architecture is splitting the input image in MXM grid and for every grid generation 2 bounding boxes and sophistication probabilities for those bounding boxes. Likely, the bounding box which represents the area of the detected object is larger than the calculated grid itself.

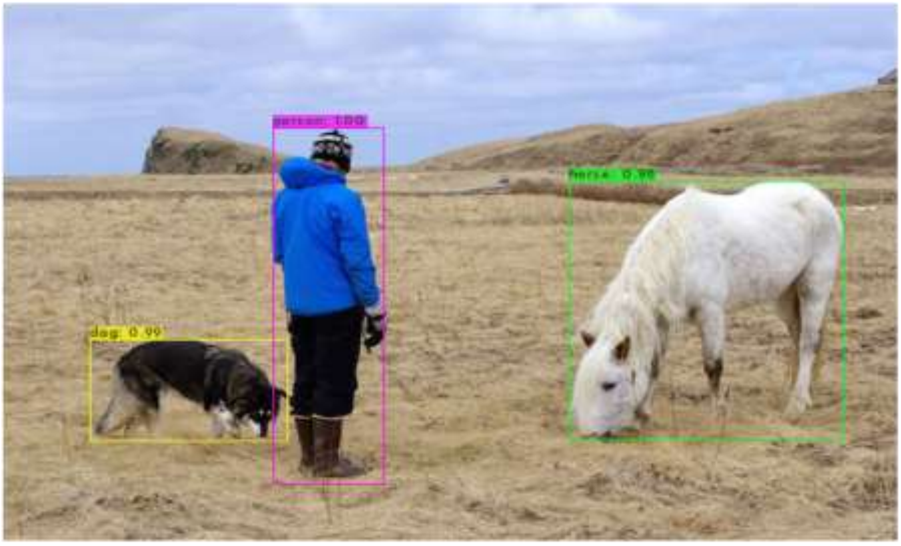


Fig 2.3.1:- Object Detection using YOLO

3. Proposed Architecture

YOLO's interface has 24 convolutional layers followed by 2 entirely connected layers. It simply uses 1 × 1 reduction layers followed by 3 × 3 convolutional layers.

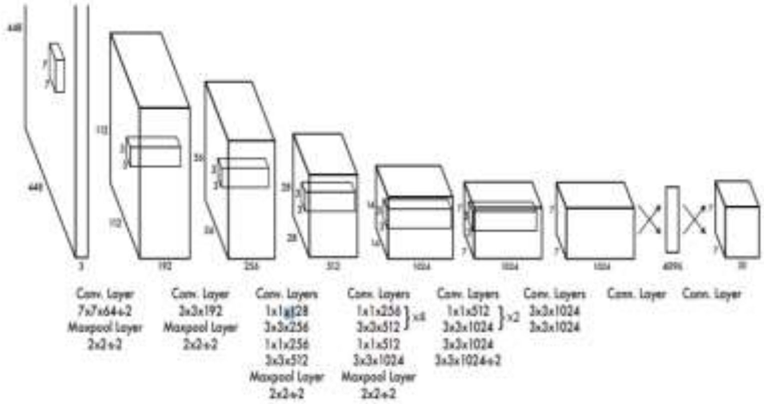


Fig 3.1.1:- The Architecture

Fast YOLO practices a neural network with 9 convolutional layers instead of 24 and fewer filters in those layers. Leaving apart the volume of the network, all training and testing parameters are the same between YOLO and Fast YOLO. YOLO is optimized for sum-squared error within the output of our model. It implements sum-squared error because it is easy to optimize, even though it doesn't align to maximize average precision. It weights localization error uniformly with classification error which is not prototypical. Also, in every image, many grid cells don't contain any object. This drives the “confidence” of many of those cells towards zero, often overwhelming the gradient from cells that do contain objects. this will cause model instability, causing training to diverge early. To change this, YOLO intensifies the loss from bounding box coordinate predictions and decreases the loss from confidence predictions for boxes that don’t contain objects. YOLO uses two parameters, λcoord and λnoobj to achieve this. YOLO sets λcoord = 5 and λnoobj = .5. The sum-squared error also equally weights errors in large boxes and small boxes. Its error metric should reflect that tiny deviation in large boxes matters but small boxes. To partially address this we predict the basis of the bounding box width and height instead of the width and height directly. YOLO predicts multiple bounding boxes per grid cell. At the time of training, we only want individual bounding box predictors to be liable for each object. We assign one predictor to be “responsible” for predicting an object supported which prediction has the very best current IOU with the bottom truth. This results in specialization between the bounding box predictors. Each predictor gets more qualified at predicting specific sizes, aspect ratios, or classes of objects, improving overall recall.

4. Implementation

4.1 Algorithm

YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding box regression
- Intersection Over Union (IOU)

Residual blocks

First, the image is divided into various grids. Each grid has a dimension of S x S. The following image shows how an input image is divided into grids.

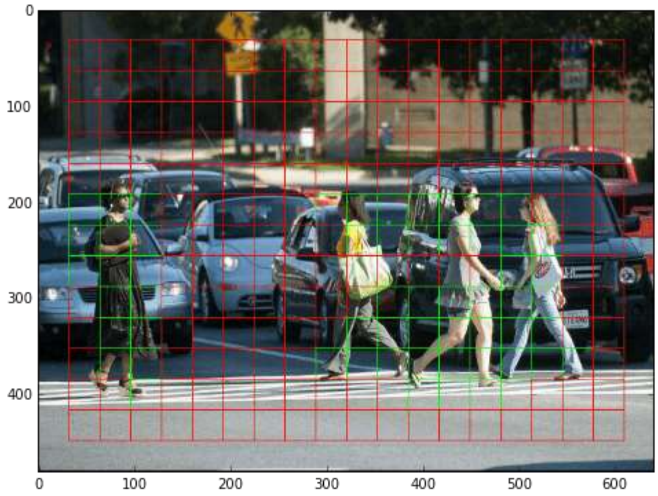


Fig 4.1.1:- Residual Blocks

In the image above, there are many grid cells of equal dimension. Every grid cell will detect objects that appear within them. For example, if an object center appears within a certain grid cell, then this cell will be responsible for detecting it.

Bounding box regression

A bounding box is an outline that highlights an object in an image.

Every bounding box in the image consists of the following attributes:

- Width (bw)
- Height (bh)
- Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
- Bounding box center (bx,by)

The following image shows an example of a bounding box. The bounding box has been represented by a yellow outline.

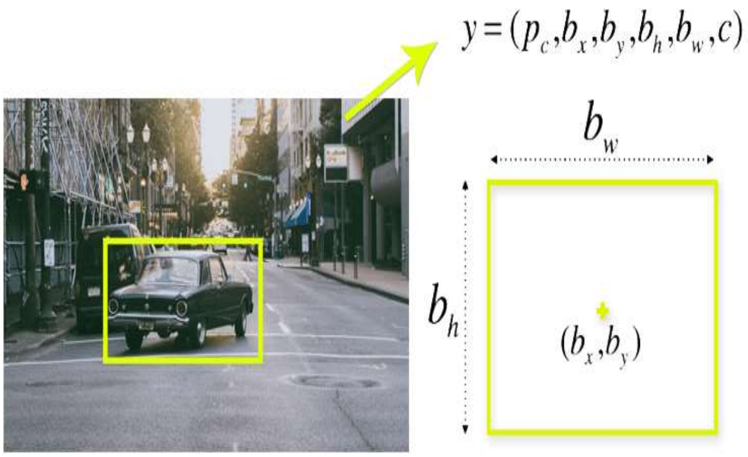


Fig 4.1.2:- Bounding Box Regression

YOLO uses a single bounding box regression to predict the height, width, center, and class of objects. In the above image, represents the probability of an object appearing in the bounding box.

Intersection over union (IOU)

Intersection over union (IOU) is a phenomenon in object detection that describes how boxes overlap. YOLO uses IOU to provide an output box that surrounds the objects perfectly.

Each grid cell is responsible for predicting the bounding boxes and their confidence scores. The IOU is equal to 1 if the predicted bounding box is the same as the real box. This mechanism eliminates bounding boxes that are not equal to the real box.

$$\text{IOU} = \frac{\text{Intersection Area}}{\text{Union Area}}$$

The following image provides a simple example of how IOU works.

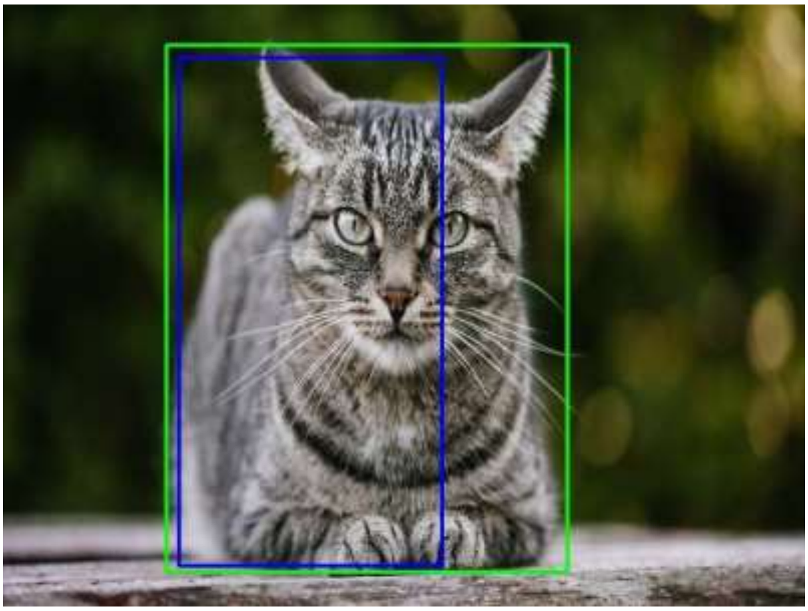


Fig 4.1.3:- Intersection Over Union

In the image above, there are two bounding boxes, one in green and the other one in blue. The blue box is the predicted box while the green box is the real box. YOLO ensures that the two bounding boxes are equal.

Combination of the three techniques

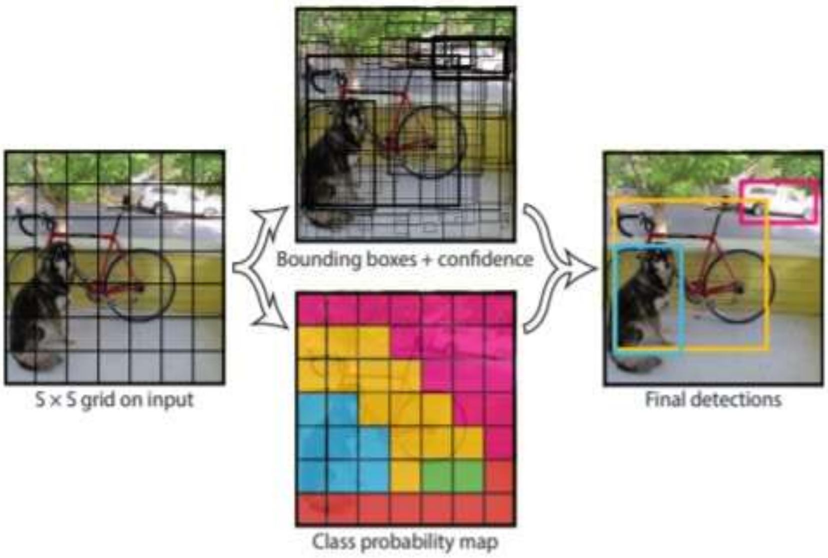


Fig 4.1.4:- The image shows how the three techniques are applied to produce the final results.

First, the image is divided into grid cells. Each grid cell forecasts B bounding boxes and provides their confidence scores. The cells predict the class probabilities to establish the class of each object.

For example, we can notice at least three classes of objects: a car, a dog, and a bicycle. All the predictions are made simultaneously using a single convolutional neural network.

Intersection over union ensures that the predicted bounding boxes are equal to the real boxes of the objects. This phenomenon eliminates unnecessary bounding boxes that do not meet the characteristics of the objects (like height and width). The final detection will consist of unique bounding boxes that fit the objects perfectly.

For example, the car is surrounded by the pink bounding box while the bicycle is surrounded by the yellow bounding box. The dog has been highlighted using the blue bounding box.

4.2 Code Implementation

Darknet: Darknet is an open-source neural network structure. It's a speedy and extremely specific, framework for real-time object detection where accuracy for custom trained model depends on training data, iterations, batch size, etc. The major reason it is quick because it is written in C and CUDA.

TensorFlow: TensorFlow is an end-to-end open-source platform for machine learning and numerical computation, the tactic of acquiring data, training models, serving predictions, and refining future results. TensorFlow brings together models and algorithms for Machine Learning and Deep Learning. It's originally based on C++ and uses Python as the front end. TensorFlow is at the present the foremost popular software library. Multiple real-world applications of deep learning and Machine Learning make TensorFlow popular. Being an Open-Source library for deep learning and machine learning, TensorFlow finds a task to play in text-based applications, image recognition, voice search, and lots more. Deep Face, Facebook's image recognition system, uses TensorFlow for image recognition. it's employed by Apple's Siri for voice recognition. Every Google app that you simply use has made good use of TensorFlow to form your experience better.

OpenCV: OpenCV supports a good sort of programming languages like C++, Python, Java, etc., and is out there on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations supported by CUDA and OpenCL also are under active development. In OpenCV, a video is often read either by using the feed from a camera connected to a computer or by reading a video file. the primary step towards reading a video file is to make a Video Capture object. Its argument is either the device index or the name of the input file to be read. In most cases, just one camera is connected to the system. So, all we do is pass '0' and OpenCV uses the sole camera attached to the pc. When quite one camera is connected to the pc, we will select the second camera bypassing '1', the third camera bypassing '2', and so on.

5. Result

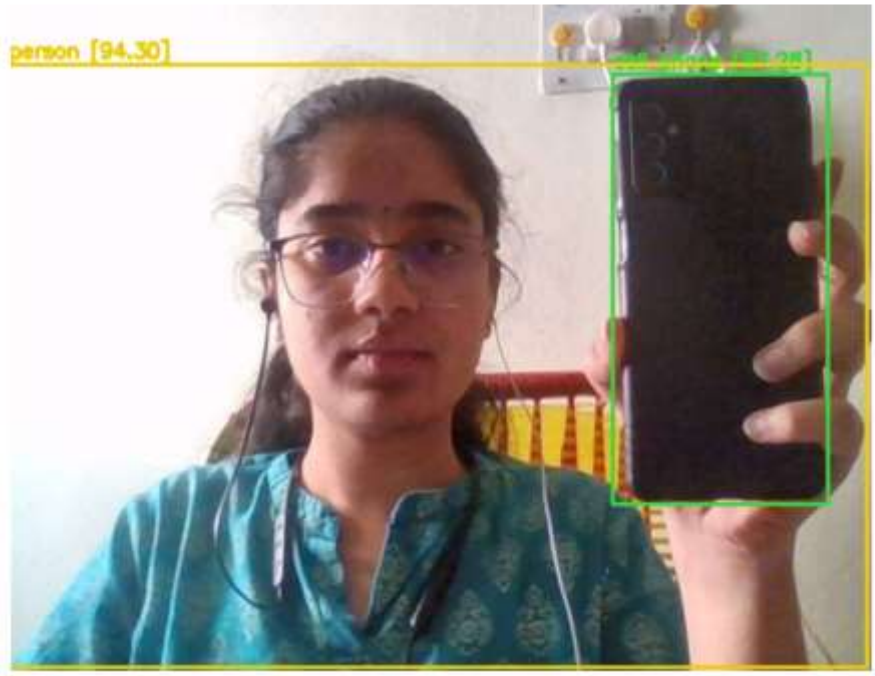


Fig 5.1:- Output Screen 1

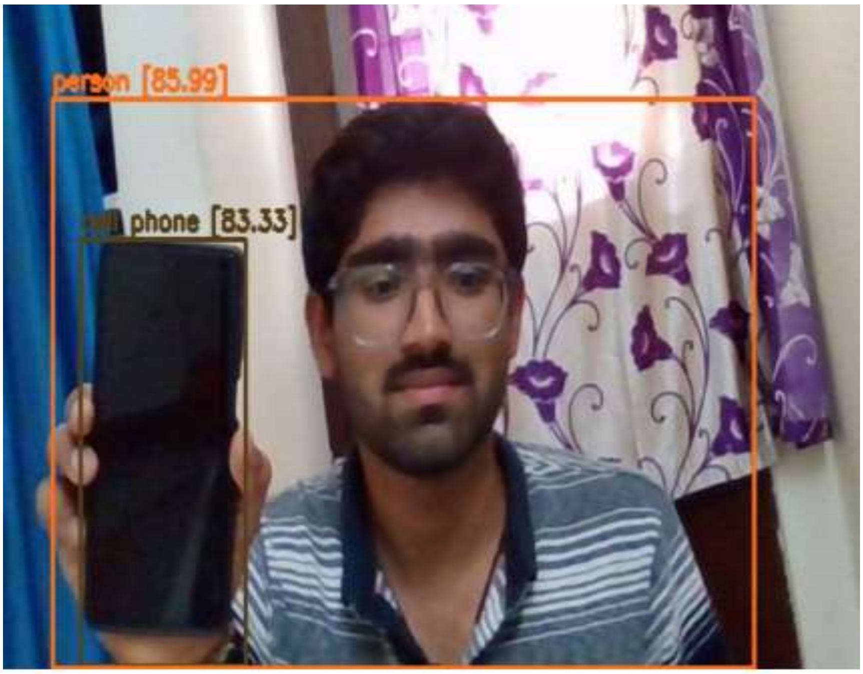


Fig 5.2:- Output Screen 2

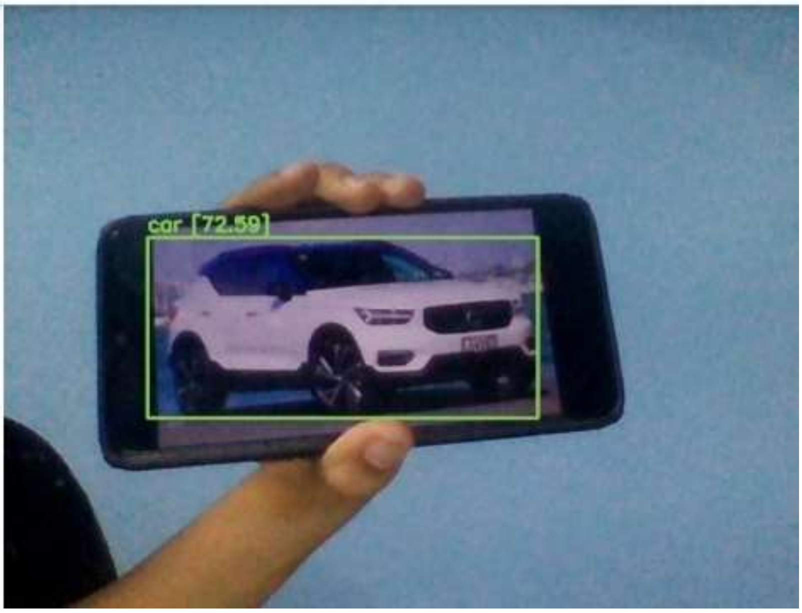
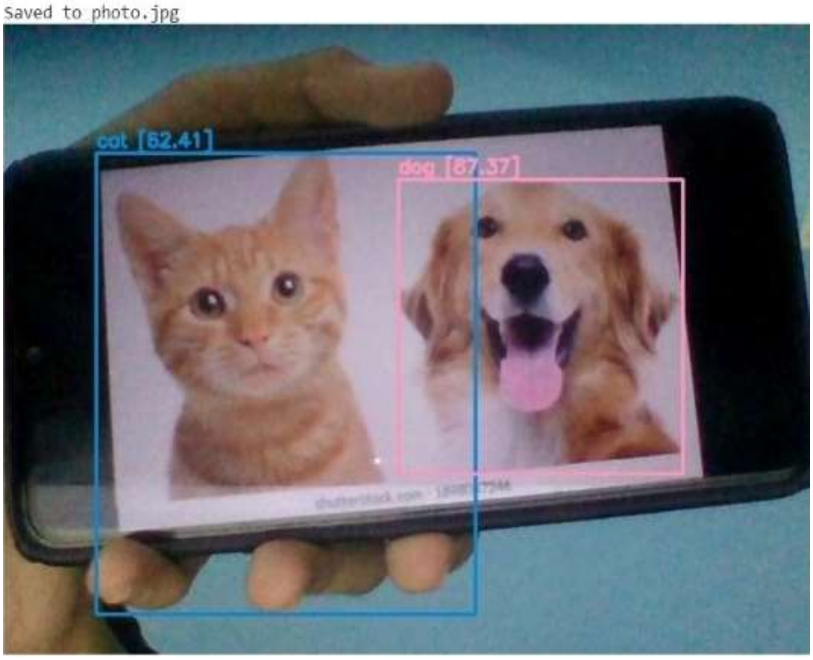


Fig 5.3:- Output Screen 3



Saved to photo.jpg

Fig 5.4:- Output Screen 4

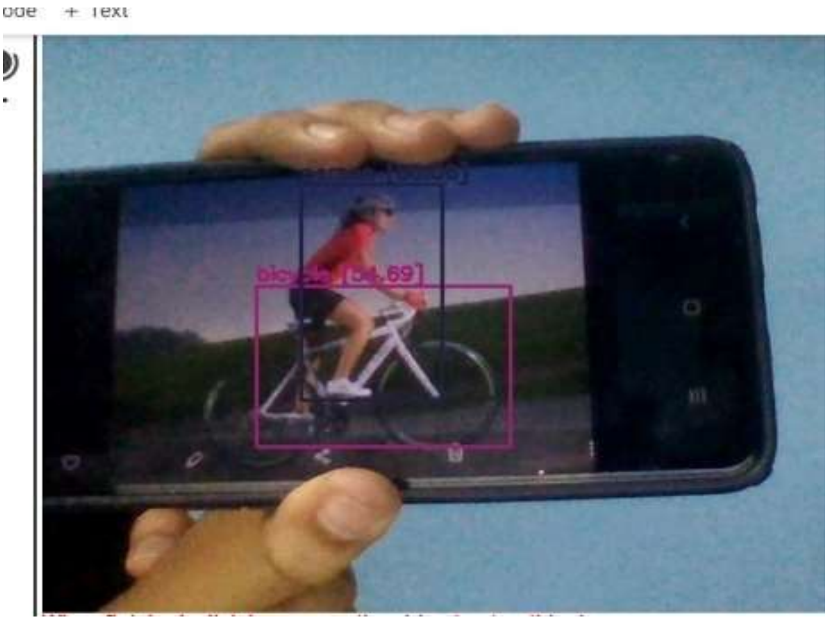


Fig 5.5:- Output Screen 5

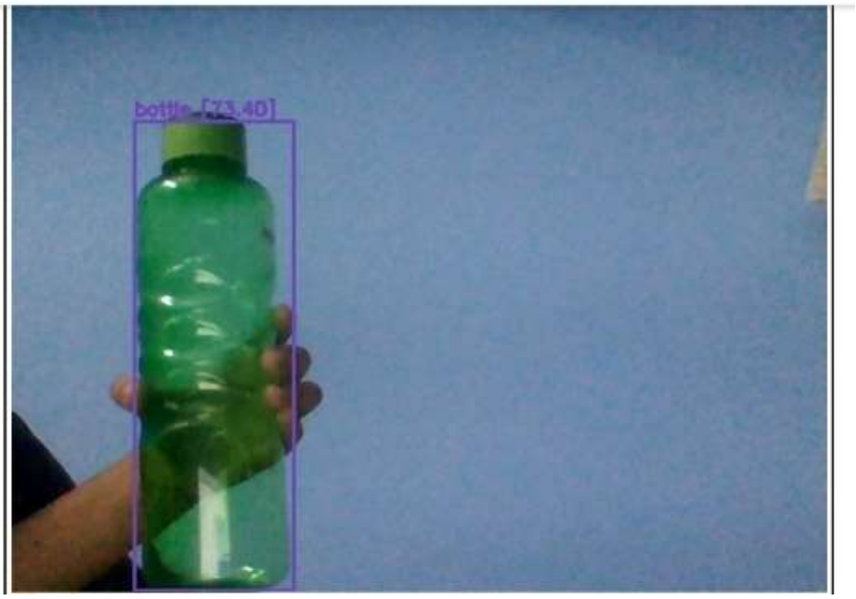


Fig 5.6:- Output Screen 6

6. Conclusion

Object detection is a key ability for most computer and robot vision system. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated in assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning. It should be noted that object detection has not been used much in many areas where it could be of great help.This approach helps in increasing the accuracy and speed and achieves the desired results. By using method, we are able to detect object more precisely and identify the objects individually with exact location of an object in the picture. Implementations of the YOLO algorithm on the web using Darknet is one open-source neural network framework. Darknet makes it really fast and provides for making computations on a GPU, essential for real-time predictions. The object detection system can be applied in the area of surveillance system, face recognition, fault detection, character recognition etc.

7. Future Scope

In future we can add a faster model that runs on the GPU and use a camera that provides a 360 field of view and allows analysis completely around the person. We can also include a Global Positioning System and allow the person to detect the objects instantly without any delay in frames and seconds. Herewith are some of the main useful applications of object detection: Vehicle’s Plates recognition, self-driving cars, Tracking objects, face recognition, medical imaging, object counting, object extraction from an image or video, person detection.As mobile robots, and in general autonomous machines, are starting to be more widely deployed (e.g., quad-copters, drones and soon service robots), the need of object detection systems is gaining more importance. Finally, we need to consider that we will need object detection systems for nano-robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical.

8. References

1. [1] Datta, R., Li, J., Wang, J.Z.: Content-based image retrieval: approaches and trends of the new age. In: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval. pp. 253–262 (2005).
2. [2] Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: Unifying structure and texture similarity. arXiv preprint arXiv:2004.07728 (2020).
3. [3] Gudivada, V.N., Raghavan, V.V.: Content based image retrieval systems. Com-
4. puter 28(9), 18–22 (1995).
5. [4] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015).
6. [5] Jung, H., Lee, S., Yim, J., Park, S., Kim, J.: Joint fine-tuning in deep neural networks for facial expression recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 2983–2991 (2015).
7. [6] Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence 40(12), 2935–2947 (2017).
8. [7] Ragkhitwetsagul,C.,Krinke,J.,Marnette,B.:Apictureisworthathousandwords: Code clone detection based on image similarity. In: 2018 IEEE 12th International workshop on software clones (IWSC). pp. 44–50. IEEE (2018).
9. [8] Shnain, N.A., Hussain, Z.M., Lu, S.F.: A feature-based structural measure: an image similarity measure for face recognition. Applied Sciences 7(8), 786 (2017).
10. [9] Wang,L.,Rajan,D.:Animagesimilaritydescriptorforclassificationtasks.Journal of Visual Communication and Image Representation 71, 102847 (2020).
11. [10] Wang, Y.X., Ramanan, D., Hebert, M.: Growing a brain: Fine-tuning by increasing model capacity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2471–2480 (2017).
12. B Subbarayudu, Srija Harshika D, E Amareswar, R Gangadhar Reddy, Kishor Kumar Reddy C, “Review and Comparison on Software Process Models”, International Journal of Mechanical Engineering and Technology, 2017.
13. (http://www.iaeme.com/MasterAdmin/UploadFolder/IJMET_08_08_105/IJMET_08_08_105.pdf)
14. Kishor Kumar Reddy C and Vijaya Babu B, “ISLGAS: Improved Supervised Learning in Quest using Gain Ratio as Attribute Selection Measure”, International Journal of Control Theory and Applications, 2016.
15. Kishor Kumar Reddy C and Vijaya Babu B “A Survey on Issues of Decision Tree and Non-Decision Tree Algorithms”, International Journal of Artificial Intelligence and Applications for Smart Devices, 2016.
16. Kishor Kumar Reddy C, Rupa C H and Vijaya Babu B, “ISLIQ: Improved Supervised Learning in Quest to Nowcast the presence of Snow/No-Snow”, WSEAS Transactions on Computers, 2016.

17. (http://www.wseas.org/multimedia/journals/computers/2016/a055705-872.pdf)
18. Kishor Kumar Reddy C and Vijaya Babu B, “ISPM: Improved Snow Prediction Model to Nowcast the Presence of Snow/No-Snow”, International Review on Computers and Software, 2015.
19. (http://www.praiseworthyprize.org/jsm/index.php?journal=irecos&page=article&op=view&path%5B%5D=17055)
20. Kishor Kumar Reddy C, Anisha P R, Narasimha Prasad L V, “A Pragmatic approach for Detecting Liver Cancer using Image Processing and Data Mining Techniques”, IEEE International Conference on Signal Processing And Communication Engineering Systems, Guntur, India, January 2015.
21. Azmathulla Shaik, Kishor Kumar Reddy C, Anisha P R and Siddarth K, “AMTS: Advanced Movie Ticketing System”, ACM International Conference on Information and Communication Technology for Competitive Strategies, Rajasthan, India, November 2014.
22. Azmathulla Shaik, Kishor Kumar Reddy C, Anisha P R and Ravi Shekar Reddy A, “MRTS: A Robust and Scalable Architecture for Metro Rail Ticketing System”, IEEE International Conference on Computational Intelligence and Communication Networks, Bhopal, India, November 2014.
23. Narasimha Prasad, Kishor Kumar Reddy and Ramya Tulasi Nirjogi, “A Novel Approach for Seismic Signal Magnitude Detection Using Haar Wavelet”, IEEE International Conference on Intelligent Systems, Modelling and Simulation, October 2014.
24. Narasimha Prasad Lakkakula, Kishor Kumar Reddy, and Murali Prasad Raja, “Remote Sensing Of Snow Wrap Using Clustering And Wavelet Transform,” IEEE International Conference on Mathematical Modelling and Computer Simulation, Malaysia, September 2014.
25. Narasimha Prasad Lakkakula, Mannava Munirathnam Naidu, and Kishor Kumar Reddy, “An Entropy Based Elegant Decision Tree Classifier to Predict Precipitation”, IEEE European Modelling Symposium, Italy, October, 2015.
26. Kishor Kumar Reddy, Anisha P R and Narasimha Prasad L V, “A Novel Approach for Detecting the Bone Cancer and its Stage based on Mean Intensity and Tumor Size”, International Conference on Recent Researchers in Applied Computer Science, 2015.
27. Kishor Kumar Reddy, Anisha P R and Narasimha Prasad L V, “Detection Of Thunderstorms Using Data Mining and Image Processing”, IEEE International Conference on the Applications of the Digital Information and Web technologies, Chennai, India, February 2014, pp. 226-231.
28. Narasimha Prasad, Kishor Kumar Reddy and Ramya Tulasi Nirjogi, “A Novel Approach for Seismic Signal Magnitude Detection Using Haar Wavelet”, IEEE International Conference on Intelligent Systems, Modelling and Simulation, Malaysia, January 2014, pp. 324-329.