JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

YOLO BASED OBJECT DETECTION

[1]Prof.Haseeba Yaseen,[2]Prof.Gadige Radha

[1]Assistant Professor,[2]Assistant Professor
[1]Department of Information Technology
[1]Vasavi College of Engineering, Hyderabad-31,India.

Abstract : Object Detection using machine learning has achieved very good performance but there are many problems with images in such as blur or rotating jitter, etc. The aim is to detect objects and real-time objects using You Only Look Once(YOLO) and YOLOv3(YOLO version 3) approach. Object detection in YOLO is done as a regression problem and provide the class probabilities of the detected images. The YOLO methodology has many benefits when compared to other object detection strategies. The biggest advantage of using YOLO is its super speed – it is incredibly fast and can process 45 frames per second. YOLOv3 is a real-time object detection algorithm that identifies specific objects in videos, live feeds or image. We have used this method for detecting various types of objects and created a web application that fetch objects from files and also through web cam to detect using flask framework.

IndexTerms **- Detection, YOLO, YOLOv3, Deep learning, Convolution Neural Network, Bounding boxes.**

I. INTRODUCTION

Object detection is a computer technique that allows us to identify and locate objects in an image or video. Specifically, object detection draws bounding boxes around these detected objects, which allows us to locate where said objects are in. Object detection algorithm is divided into traditional machine learning approaches where computer vision techniques are used to look at various features of an image, such as the color, histogram or edges, to identify groups of pixels that may belong to an object. These features are then fed into a regression model that predicts the location of the object along with its label. In this, our main objective is to detect multiple objects from an image and objects in live streams. There are several different techniques for object detection, they can be divided into two categories, first is the algorithms based on classifications. The second category is the algorithms based on regressions. YOLO methods comes under second category.

The YOLO algorithm first separates an image into grid. Each grid cell predicts some number of boundary boxes around objects that score highly with the aforementioned predefined classes. In simple words, we predict the classes and boundary boxes of the whole image at a single run of the algorithm and then it is going to detect multiple image objects using a single neural network. The YOLO algorithm is faster when compared to other algorithms. The main disadvantage of YOLO is low recall and more localization error.

II. EXSISTINGSYSTEM

Object Detection and Classification using YOLOv3 paper written by Dr. S. V Viraktamath. The prior work is to detect objects using deep learning techniques such as the Faster Regional Convolution Neural Network(F-RCNN), the You Only Look Once (YOLO), the Single Shot Detector (SSD). To improve precision of object detection they have used YOLO method in this paper [1]. YOLO based Detection and Classification of Objects in video records paper written by Arka Prava Jana. In this paper, they have generalized the error while detecting objects in video records and used YOLOv2 to detect real-time objects and improve the computation and processing speed [2].

You Only Look Once: Unified, Real-Time Object Detection, paper by Joseph Redmon. Their work is on detecting objects using a regression algorithm. To get higher accuracy and predictions they have proposed YOLO model in this paper [3]. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers paper written by Rachel Huang. This paper mainly focuses on YOLO-LITE, a real-time object detection model to run on portable devices such as a laptop or cellphone lacking a Graphics Processing Unit (GPU) [4]. Object Detection in Shelf Images with YOLO by Ceren Gulra Melek. In this paper, their aim to detect objects in shelf images which solves many problems such as monitoring the number of products on the shelves, completing the missing products and matching the planogram continuously [5].

In existing YOLO model it detects the object with the help of convolution neural network and bounding boxes. The image or the dataset is first bounded around the box and it is detected by the datasets provided in the digital format. The dataset consists of different images in which the object has to be detected. In context to the speed and accuracy are not up to mark and also the object detected is only from the dataset which is provided by the user. The dataset can have as many images that the user wants to have. And object detected from the dataset may end with less accuracy and also the execution speed of the model is also slow. However, the model yields with the good accuracy and detects the object with constant speed. The image detected can have more than one object in it. The model can detect multiple objects at same time and also with good accuracy. The model developed detects the objects based on the dataset provided by the user with consistency. The issues with the developed model are that speed of the model is low, accuracy of the developed model is less and also the model developed cannot detect the objects with the help of the webcam.

III. PROPOSED SYSTEM

To address the above issues, a new efficient algorithm named YOLOv3 is designed to develop the existing model. The newly proposed model yields with high accuracy, the proposed model speed is increased and also the model can detect the object shown in webcam. The model can directly detect the object via webcam itself and also it can detect the object in the images as well. This makes the model more users friendly and also it is more efficient. Also, the model developed can perform the tasks of detecting the object within fraction of seconds with high accuracy. The proposed model can have various dataset provided by the user and also can detect the objects based on the dataset provided. With the help of the developed model the user can experience an user-friendly model and it also helps the user to detect the object with least amount of time with high accuracy than the one which has no webcam function in the developed model.

IV. SYSTEM DESIGN AND METHODOLOGY

4.1. HARDWARE COMPONENTS

1) PROCESSOR : Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
2) RAM : 8.00 GB

4.2. SOFTWARE COMPONENTS

1) OPERATING SYSTEM : Windows XP/10
2) CODING LANGUAGE : Python
3) TOOLS : Flask framework, Python

The aim is to detect objects and real-time objects using You Only Look Once (YOLO) and YOLOv3(YOLO version 3) approach. In this paper, we are creating a web application where the objects are fetched and detected through web using flask framework. Flask is an API of Python that allows us to build up web applications. YOLO algorithm uses a completely different approach. The algorithm applies a single neural network to the entire full image. Then this network divides that image into regions which provides the bounding boxes and also predicts probabilities for each region. These generated bounding boxes are weighted by the predicted probabilities.

4.3. WORK FLOW OF YOLOv3

YOLO is a convolution neural network. It consists of a total of 24 convolutional layers and followed by 2 fully connected layers. Each layer has its own importance and the layers are separated by their functionality. The First 20 convolutional layers followed by an average pooling layer and a fully connected layer is pre-trained on the ImageNet dataset which is a 1000-class classification dataset. The pretraining for classification is performed on the dataset with the image resolution of 224 x 224×3. The layers comprise 3×3 convolutional layers and1x1 reduction layers. For object detection, in the end, the last 4 convolutional layers followed by 2 fully connected layers are added to train the network. Object detection requires more precise detail hence the resolution of the dataset is increased to 448 x 448. Then the final layer predicts the class probabilities and bounding boxes. All the other convolutional layers use leaky ReLU activation whereas the final layer uses a linear activation. The input is of 448 x 448 image and the output is the class prediction of the detected object enclosed in the bounding box.

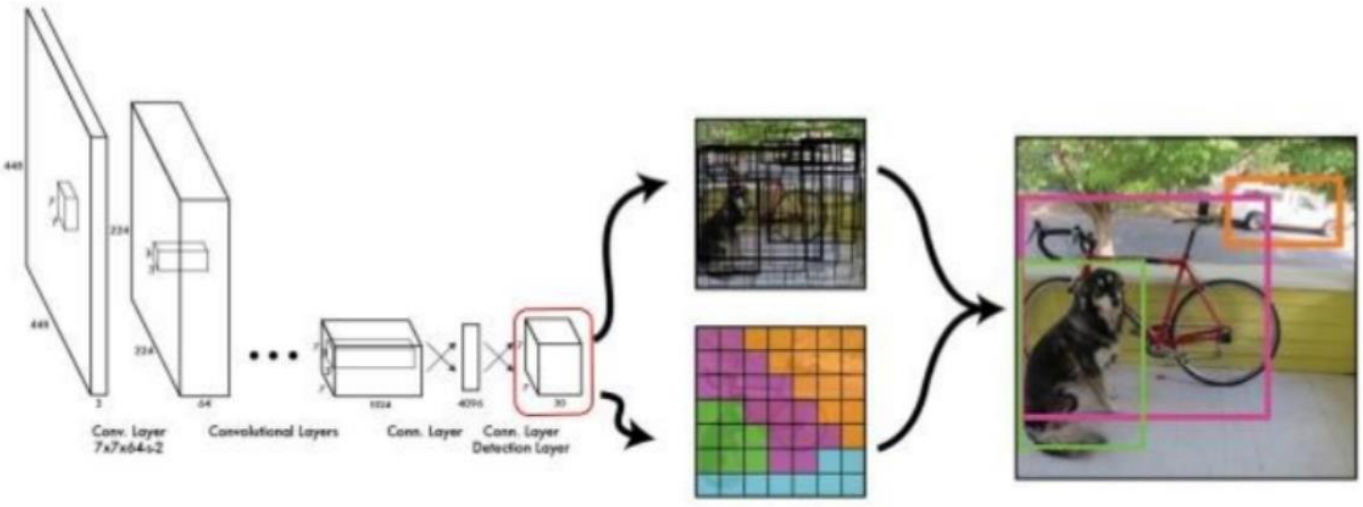

Fig 1. Working flow of YOLOv3.

The web interface of the proposed home automationsystem is used to operate and monitor the state of various appliances and user commands that can be easily modified to support additional hardware interface modules.

V. IMPLEMENTATION

First, it divides the image into a 13×13 grid of cells. The size of these 169 cells vary depending on the size of the input. For a 416×416 input size that we used in our experiments, the cell size was 32×32. Each cell is then responsible for predicting a number of boxes in the image. For each bounding box, the network also predicts the confidence that the bounding box actually encloses an object, and the probability of the enclosed object being a particular class. Most of these bounding boxes are eliminated because their confidence is low or because they are enclosing the same object as another bounding box with very high confidence score. This technique is called non-maximum suppression. The YOLOv3 is faster and more accurate than their previous work YOLOv2. YOLOv3 handles multiple scales better. They have also improved the network by making it bigger and taking it towards residual networks by adding shortcut connections.

Fig 2. Work of YOLO

5.1. PREDICTION OF BOUNDINGBOX

our system predicts bounding boxes using dimension clusters as anchor boxes. The network predicts 4 coordinates for each bounding box, tx, ty, tw, th. If the cell is offset from the top left corner of the image by (cx, cy) and the bounding box prior has width and height pw, ph, then the predictions correspond to: bx = σ(tx) + cx by = σ(ty) + cy bw = pwe tw bh = phe th. During training we use sum of squared error loss. If the ground truth for some coordinate prediction is tˆ * our gradient is the ground truth value minus our prediction: tˆ * − t* . This ground truth value can be easily computed by inverting the equations above. YOLOv3 predicts an objectness score for each bounding box using logistic regression. This should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior.

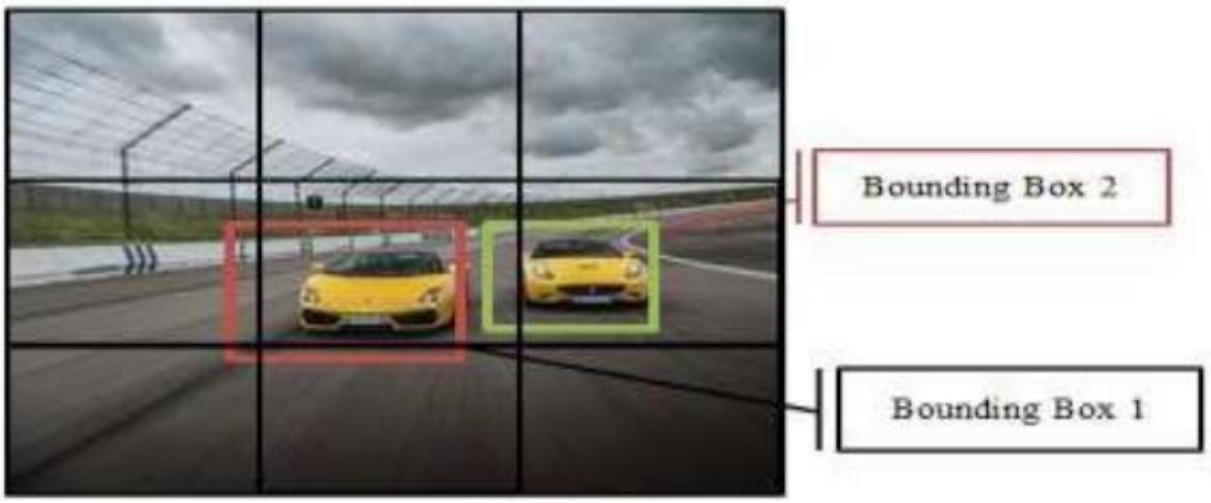

Fig 2.1.Bounding boxes

5.2. PREDICTION OF CLASSES

Each box predicts the classes the bounding box may contain using multilabel classification. We use independent logistic classifiers. During training we use binary cross-entropy loss for the class predictions. This formulation helps when we move to more complex domains like the Open Images Dataset. In this dataset there are many overlapping labels. A multilabel approach better models the data.

YOLOv3 predicts boxes at 3 different scales. Our system extracts features from those scales using a similar concept to feature pyramid networks. From our base feature extractor it add several convolutional layers. The last of these predicts a 3-d tensor encoding bounding box, object, and class predictions. We predict 3 boxes at each scale so the tensor is N × N × [3 ∗ (4 + 1 + 80)] for the 4 bounding box offsets, 1 object prediction, and 80 class predictions. Next it takes the feature map from 2 layers previous and up-sample it by 2×. It also take a feature map from earlier in the network and merge it with our up-sampled features using concatenation. This method allows us to get more meaningful semantic information from the up-sampled features and finer-grained information from the earlier feature map. We then add a few more convolutional layers to process this combined feature map, and eventually predict a similar tensor, although now twice the size. We perform the same design one more time to predict boxes for the final scale.

VI. Results

We explore the best approach to accomplish the image to detect the object. We use python in the back-end and HTML and CSS for front-end. The model developed can detect the objects from the image as well as directly via webcam with high accuracy and fast. The detected image may contain multiple objects in it.

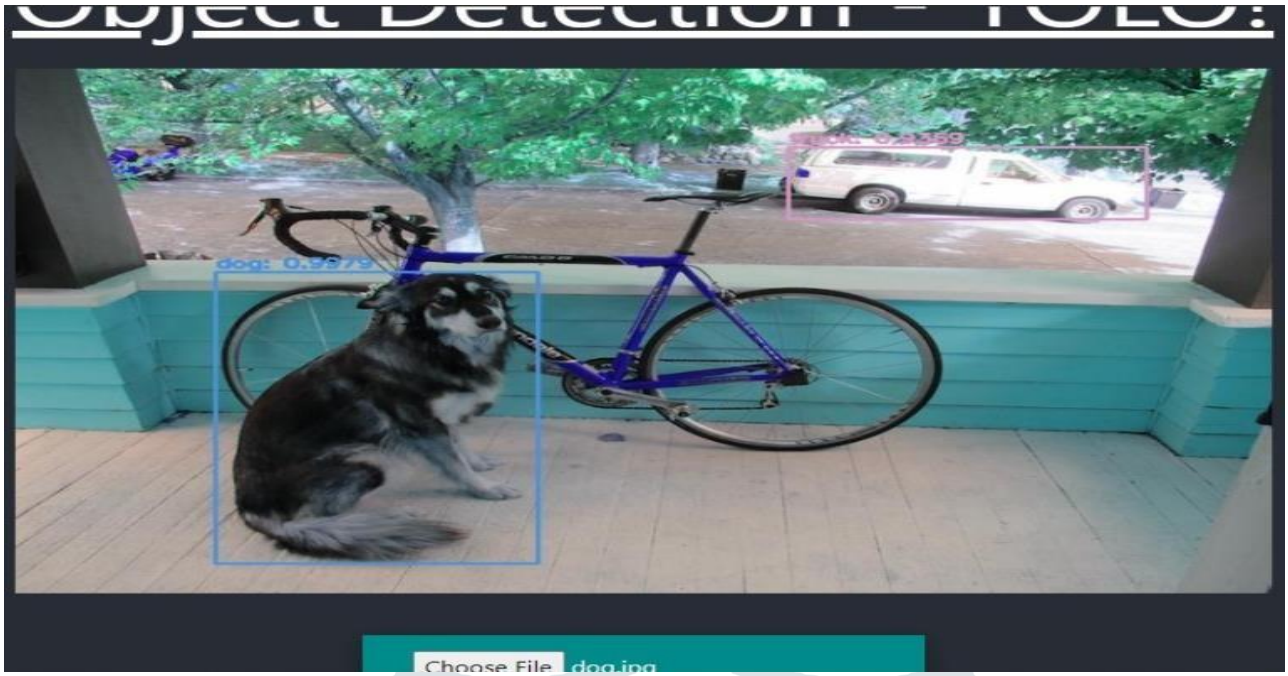

Fig 3. Object detected from image

The above figure describes the object detected with the help of dataset of the object in the digital format. Firstly the image is chosen from the dataset provided by the user with the help of the image from the dataset the object present inside the image is detected with the help of YOLOv3 algorithm.

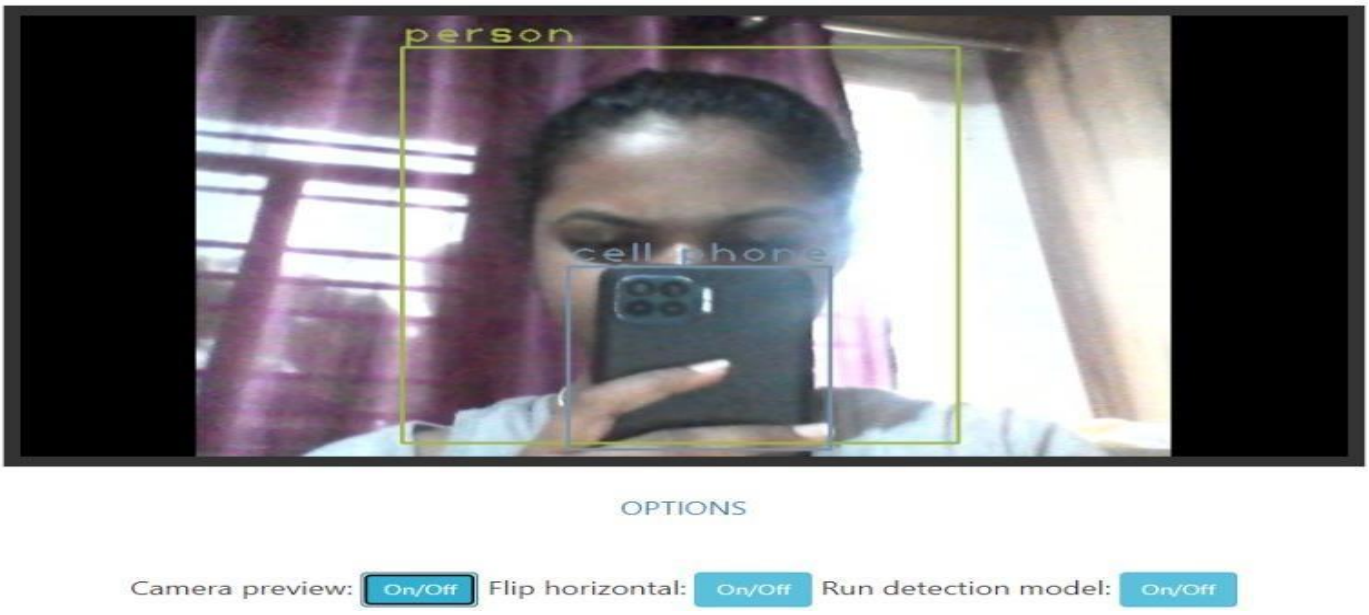

Fig 4. Object detected from webcam

The above figure describes the object detected with the help of the webcam as a tool to detect. Here the object detected is directly via webcam and also it detects multiple objects with the help of webcam. Here in this model webcam acts as a tool to detect the objects.

VII. CONCLUSION

In this paper, we have used YOLO algorithm for object detection also flask framework because of its advantages. The YOLO algorithm can be implemented in different fields to solve many real-time problems such as monitoring traffic rules, security etc. Here, we proposed a user-friendly application to detect the objects in the image as well as with the help of webcam. The result of the model are in term of the speed of the model and also with the accuracy of the model. Our proposed model has high accuracy and also the speed of the model increased. Hence, the model developed is of high efficiency. By concluding we can say that our developed YOLO object detection model yields high accuracy with increase in speed and also can detect the objects not only with the datasets provided but also with webcam through flask framework

VIII. REFERENCES

[1] Analogy. Wikipedia, Mar 2018. 1

[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303– 338, 2010. 6

[3] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017. 3

[4] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. Iqa: Visual question answering in interactive environments. arXiv preprint arXiv:1712.03316, 2017. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 3

[6] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. 3

[7] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Openimages: A public dataset for largescale multi-label and multi-class image classification. Dataset available from https://github.com/openimages, 2017. 2

[8] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017. 2, 3

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. ´ Focal loss for dense object detection. arXiv preprint arXiv:1708.02002, 2017. 1, 3, 4

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com- ´ mon objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 2

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.- Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016. 3

[12] I. Newton. Philosophiae naturalis principia mathematica. William Dawson & Sons Ltd., London, 1687. 1

[13] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. Rubenstein. Animal population censusing at scale with citizen science and photographic identification. 2017. 4

[14] J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016. 3

[15] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017. 1, 2, 3

[16] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018. 4

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015. 2