IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Object Detection From Multimedia For Automated Systems

[1]Pawan Singh, [2]Sarthak Naik, [3]Ishita Patri and [4] Prof. Jinesh Melvin

[123]Student, [4]Head of Department
Department of Information Technology
Pillai College of Engineering, University of Mumbai, New Panvel, India

Abstract: Computer vision and image detection using deep learning and AI software systems which can recognize as well as understand images and scenes has really become a useful and important aspect for the modern and future AI projects and development. Object detection is widely used for vehicle detection, pedestrian counting, web images, traffic analyzing systems and self-driving cars. In this project, we are using highly accurate object detection-algorithms and methods such as YOLOv5, RetinaNet based on TensorFlow algorithms and keras libraries. The tensorFlow and openCV based detection and classification uses GPU to increase the computation speed and processes . Using these methods and algorithms, based on machine learning, require lots of mathematical and deep learning framework understanding by using dependencies such as TensorFlow, OpenCV. We are proposing a system to detect each and every object in image by the area object in highlighted rectangular boxes and identify each and every object and assign its tag to the object. This also includes the precision of each and every method for identifying objects.

***Index Terms* - YOLO- You Only Look Once ,Faster R-CNN, Keras, Tensorflow.**

I.INTRODUCTION

Humans look at an image and instantly process the objects in it and determine their locations due to the interlinked neurons of the brain. Just like the human interpretation, the world today requires fast and accurate algorithms to classify and detect various objects for various applications. Recognition, as it achieved a large decrease in error rate but at the expense of speed and computation time. YOLO("You Only Look Once"), OPENCV, PYTORCH,COCO dataset, TKINTER with MYSQL(MySQL is optional),GPU are the methodology used to detect, count and track the objects in MOT.The proposed system uses the Latest YoloV5 which is used to detect the objects.YoloV5 uses pytorch classifier for training as well as detection. Yolo begins its journey with darknet technology ,which was later developed to yolov2 ,then yolo v3 and later to yolo v4.And now for easy building of object detection yolo v5 was introduced leading to better performance of object detection .Yolo V5 is constructed with Pytorch Classifier in deep learning and after object detection the opencv module is used for inputting real-time or file format video input to the algorithm and also tracks, and counts the objects detected in the output obtained making the system an efficient system.

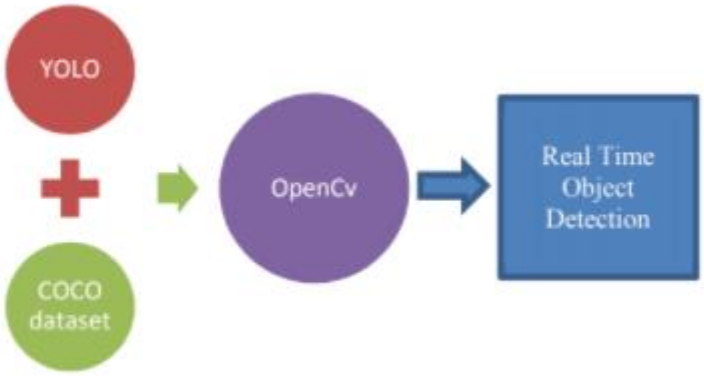


Fig. 3. Architecture of CPU Based YOLO

II. LITERATURE SURVEY

In this chapter the relevant techniques in literature are reviewed. It describes various techniques used in the work. Identify the current literature on related domain problems. Identify the techniques that have been developed and present the various advantages and limitations of these methods used extensively in literature. Object detection is the identification of an object in the image along with its localisation and classification. It has widespread applications and is a critical component for vision based software systems. A salient feature point based algorithm for multiple object tracking in the presence of partial object occlusion has been proposed in. In this method, extract the prominent feature points from each target object and then use a particle filter based approach to track the feature points in image sequences based on various attributes such as location, velocity and other descriptors. The evaluation in computer vision with Deep Learning has been established and accomplished with time, primarily over one particular well-known algorithm named Convolutional Neural Networks (CNN).Most of the CNN based detection methods for example R-CNN], starts by recommending different locations and scales present in a test image as a input to the classifiers of objects, at the point of training and return the classifiers of resultant proposed region to detect an object.A major challenge in many of the object detection systems is the dependency on other computer vision techniques for helping the deep learning- based approach, which leads to slow and non-optimal performance.Techniques that are entirely computer based mostly need vast quantities of GPU power and even then aren't always real time, making them inappropriate for day to day applications.The machine learning solutions revolve around data gathering, training a model, and use the trained model to make predictions.

2.1.YOLO — You Only Look Once: 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 have high probabilities of containing the object. YOLO or In YOLO a single convolutional network predicts the bounding boxes and the class probabilities for these boxes.

YOLO V5 The original author of the yolo algorithm is Joseph RedMon. When he started the yolo algorithm construction and when it didn't have significant progress in another author Alexey Bochkovskiy published a paper on yolo and then after that a series of yolo arrived which led to yolov2,yolov3 and then upto yolov4. In the heat of yolov4 the Ultralytics LLC team on may 30,2020 issued YOLOV5.The Yolov5 took a move from darknet to pytorch achieving 140 FPS in Tesla P100 where as in yolov4 only 50 FPS.Yolo V5 has the same advantages and has almost similar architecture as yolo v4.Yet Yolov5 makes it convenient to train and detect objects compared to yolov4.

2.2.Tensorflow: Tensorflow is an open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks,etc.. It is used for both research and production by Google. Tensorflow is developed by the Google Brain team for internal Google use. It is released under the

2.3.COCO: Dataset COCO dataset is a scope object recognition, segmentation, and captioning dataset. It begins with object segmentation where image division takes place in order to discover image boundaries and objects in it. It is used for labeling using bounding boxes in image. After that in recognition in context, a basic correlation architecture is represented between the image and the objects in it. After this we collect or gather the same type of color or gray levels. This is super pixel stuff segmentation where they help in featuring important areas and also they can reduce the input element for calculations.

2.4.OpenCV: Opencv is an open source library for computer vision modules like image processing, camera access,etc.Now Gpu is also included in the Opencv module which is an essential element for pytorch .Opencv technology developed in a fast rate and supports many algorithms to make the algorithm efficient,especially in image processing field. Opencv in python supports libraries like numpy, matplot,utilis,etc.

Opencv does few applications like video image stitching, navigation, Medical analysis, etc

III. PROPOSED WORK:

The system begins with a login page and then after optioning, taking video or providing recorded video as input to the system of object detection. Then the inputted data is processed into frames for detection. During detection the The Yolo V5 uses the Model or dataset to detect the object in the input data. By using the model after detecting objects the detected objects are classified and represented by using labels and bounding boxes around the detected objects and then it is processed into output video format.

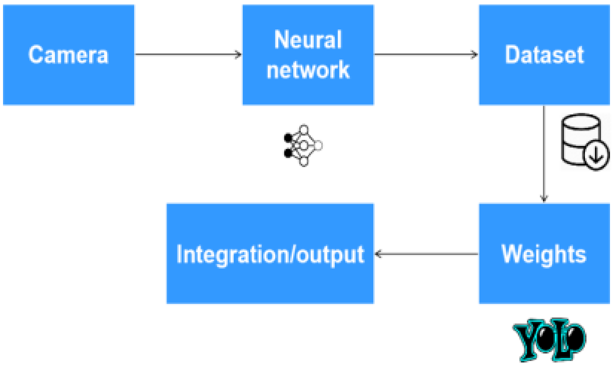


Fig. 2. General working system

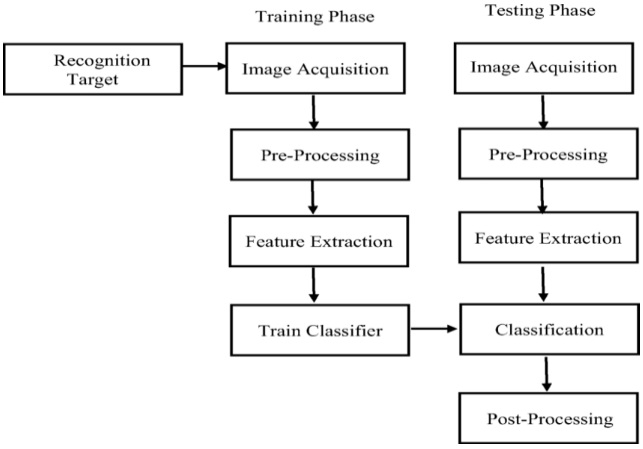
3.1.Neural Network(YOLO V5): Here using the input received from the camera or video the input data is classified into frames and each frame is sent to a yolo detection algorithm with the model which the user selected. The model can be a predefined model that is, COCO dataset model or we can create custom models for detection. Once the detection is done, it is bound with boxes and labels where the object is found and is sent to the output section where the detected frames are collected and are then compressed into output format. Before merging, the detected frames are used for tracking, counting and sorting using OpenCV and also for better results DeepSORT is also used for sorting and tracking of objects.

3.2.Dataset: This field is used for creating a custom dataset from raw images in order for creating a custom model which can be used for detection. For this the first thing is used to collect the raw images from various sources and create a dataset. Then from the dataset images the objects must be annotated and labeled from the images. For this Python frameworks like “Labeling” are used for annotating and labeling the objects. Once the annotating and labeling is done then the dataset is split into train and test images in percentage of 70% for train and 30% for test as it is the general ideal percentage used for training. Once this is done it can be sent to the yolo training algorithm where the dataset can be trained and the model can be created using the COCO dataset model.

3.3.Weights / Model: Here the labeled dataset obtained from the framework must be configured with “.yaml” (YAML Ain't Markup Language) extension format file which can be used to append the text label to the algorithm. Once the YAML file is configured it is set up in algorithm and using pytorch the given dataset gets trained using GPU according to the epochs given in algorithm for training and with test dataset the testing of trained model after completion also takes place, predicting the objects in test image. Once the objects are predicted the model is compressed into the yolo model format which is configured using the pretrained model that is using COCO dataset model. Once this is done the model file with the corresponding test result potted in graphs and texts are written in the output folder where the evaluation and testing of the model can be done by using it in a detection algorithm.

IV. System Architecture

Algorithms based on classification – they work in two stages. In the first step, we’re selecting from the image regions. Then we’re classifying those regions using convolutional neural networks.



Algorithms based on regression – ins

tead of selecting interesting parts of an image, we’re predicting classes and bounding boxes for the whole image in one run of the algorithm. Most known example of this type of algorithm is YOLO (You only look once) commonly used for real-time object detection.

V. RESULTS AND PERFORMANCE EVALUATION

The performance of YOLO is measured using mainly three terms: mAP, Precision, Recall [7]. mAP is a measure that combines recall and precision for detecting the accuracy of the object, Where it is calculated with the average precision value for recall value over 0 to 1 with IOU that is intersection over union from 0.5 to 0.95.

$$AP = \frac{\Sigma_{r=1}^{R} P_r}{R} \qquad (1)$$

$$mAP = \frac{1}{N} \Sigma \, AP_k \qquad (2)$$

Precision is used to measure how accurately the objects are predicted. Precision can describe how good the model is at predicting the positive class

$$Recall = \frac{TP}{(TP + FN)} \qquad (4)$$

IOU stands for Intersection Over Union. IoU is used to calculate between two boundaries of an image to check whether they have overlapped or not,if so, to know at what rate it is we use IoU. we will be predefining an IoU threshold (say 0.5 in our case) in order to find whether the detection was true positive or false positive.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \qquad (5)$$

The Yolo Accuracy is found using mAP. In COCO mAP, a 101-point interpolated AP definition is used in the calculation. AP corresponds to the average AP for IoU from 0.5 to 0.95 with a step size of 0.05. Consider the custom dataset model of a single class, say key, with a dataset of 200 images which was trained with the COCO dataset with the “yolov5m6.pt” model which has 51-52 GFLOPS.It was trained under 200 epochs using Pytorch Classifier using Tesla T4 GPU.The results of the main three terms discussed above were plotted in a graph as shown below in Fig. 4. The results clearly define that at a range of 0-100 images the prediction was growing and mAP was increasing and while reaching the last images the mAP was almost near to one say for the sample dataset key the Accuracy or mAP is approximately 95.39% while training as the Confidence threshold value given was 0.001.Thus we were able to train a good custom yolo model which can be used for detection. The model quality is represented using the PR curve shown below in Fig. 5.While using this model for detection, using this data the tracked object in each frame was occurring. The detected object from the crowd had mAP values from 0.2-0.9 according to the clarity of the image in real time.

VI. CONCLUSION:

In this work, we proposed an object detection and classifiction system using yolov5 that can detect objects which were trained and also they can track and take count of objects in each frame. This system has various real-time applications like detecting particular objects from object crowded environments, tracking a particular type of object or detecting a set of object classes or counting a particular object. The MOT is efficient enough to detect objects even on CPU GPU. This requires a local system with GPU which may not be affordable to all systems but with the help of free cloud sources like Google Collab we can use GPU and also make a working system which can custom train objects with raw-images. And can be used in our local systems. Therefore using the yolov5 algorithm which uses pytorch. The system is capable of tracking various objects which were trained. As a sample the system was trained with a class of object key with 200 images as dataset and after 200 epochs the dataset formed a yolo model with 95.39% mAP which was used for detection .While the custom model was used the object detection and tracking had an accuracy prediction in range of 20-90% according to the clarity of image and appearance of object in image. As this was done in CPU GPU the prediction was better than expected but was not effective as GPU as GPU system provided a better and accurate prediction of the object this is required to process a model which has 50+GFLOP data. Thus excluding that limitation. Using Deep Learning and yolov5 is executable according to user needs.

ACKNOWLEDGMENT

We would like to thank our Guide Prof. Jinesh Melvin, who gave us this golden opportunity to do this wonderful project on the topic “Object Detection from multimedia for automated systems”, which also helped us in doing a lot of Research and thereby increasing our knowledge.

We would like to express our gratitude to our H.O.D of Information Technology Dr. Satish Kumar Verma for giving us this opportunity and for motivating us to do innovative things that will be beneficial for our future.

References

[1] *Ullah, M. B., & Ullah, M. B. (2020)*. CPU Based YOLO: A Real Time Object Detection Algorithm. 2020 IEEE Region 10 Symposium (TENSYMP).

[2] *Feng, D., Haase-Schutz, C., Rosenbaum, L., Hertlein, H., Glaser, C., Timm, F., … Dietmayer, K. (2020)*. Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. IEEE Transactions on Intelligent Transportation Systems, 1–20. doi:10.1109/tits.2020.2972974

[3] *Hurtik, P., Molek, V., & Vlasanek, P. (2020).* YOLO-ASC: You Only Look Once And See Contours. 2020 International Joint Conference on Neural Networks (IJCNN). doi:10.1109/ijcnn48605.2020.9207223

[4] *Kanimozhi, S., Gayathri, G., & Mala, T. (2019)*. Multiple Real-time object identification using Single shot Multibox detection. 2019 International Conference on Computational Intelligence in Data Science (ICCIDS). doi:10.1109/iccids.2019.8862041

[4] *Abbas, S. M., & Singh, D. S. N. (2018).* Region-based Object Detection and Classification using Faster R-CNN. 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT). doi:10.1109/ciact.2018.8480413

[5] *Al-Shatnawi, M., Movahedi, V., Asif, A., & An, A. (2018).* Improving Real-Time Pedestrian Detection Using Adaptive Confidence Thresholding and Inter-Frame Correlation. 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP). doi:10.1109/mmsp.2018.8547103