Real Time Localized Multi Object Detection System

K C. Sakthi Siva Parvathi, S. Supraja Arthi, R. Yuvasree and T. Sethukarasi

Department of Computer Science and Engineering, R.M.K Engineering College, Chennai 601206, India

Object detection is a key ability required by most computer and robot vision systems. The latest research on this area has been making a great progress in many directions. Our project focuses on building a flutter based mobile application with elegant UI by implementing some popular object detection algorithms like SSD, YOLO, MobileNet and PoseNet. We use the existing real time dataset from Google's Open Image datasets,COCO, DUTS, PASCAL. To mark the performance of our project, we train our models with Google Colab's GPU and TensorFlow's object detection API. The trained models are then converted into lightweight TensorFlow Lite files and are embedded into our project directory. We incorporate a camera plugin that allows the camera in our mobile to lively capture ongoing events and based on the algorithm selected, it marks a bounding box around the objects and labels them with a high level of accuracy. By this way, our research project makes it easy for common people to use it on a day to day basis and trace out hidden or suspicious objects in their surroundings.

Keywords: Object detection, Human Pose Recognition, Tensorflow, Labeling Objects, Feature extraction

INTRODUCTION

Object detection has always been an interesting problem in the field of deep learning. Since these problems are meta-heuristic, despite a lot of research, dynamic object detection methods are still unavailable. Iterating over the problem of localization and classification we acknowledged the need for detecting and classifying multiple objects at the same time. So, we propose a solution to detect multiple objects at real-time. The goal of our object detection project is to recognize instances of a predefined set of object classes (e.g. {people, cars, bikes, animals}) and describe the locations of each detected object in the image using a bounding box.

Flutter integrated with Firebase facilatates to perform our project to a greater extent. Flutter is a UI toolkit for building native applications for mobile and web apps. Firebase enables to add datasets of various choices. This paper has the following structure. In section 2, we address the completed research works in this field. Section 3 includes information about the dataset used. Section 4 and 5 widely explains the methods we incorporate and the architecture diagram respectively. Section 6 demonstrates the steps adopted in completing the project. We also provided the results and future works found by evaluating our modals performance in section 7.

2. LITERATURE SURVEY

A detailed survey is taken under the domain of "Computer vision". In this study, we have come to know that most of them have concentrated on object detection and to count objects in a scene Ref. [6, 23], determine and track people's behaviour Ref. [8, 14, 19] and their precise locations, all while accurately labeling them.

2.1 Challenges in smaller objects detection

Object detection is customarily considered to be much harder than image classification, because of the significant challenges that still persist. Though many object detection obstacles have seen creative solutions and researchers have dedicated much effort to overcome these difficulties, amazing results are not completely achieved. All object detection frameworks continue to struggle with small objects, especially those bunched together with partial occlusions Ref. [12, 13]. There are a few challenges like dual priorities, speed, multiple scales, limited data, and class imbalance that makes object detection ineffective.

2.2 Object detection using KNN

Object detector models have gone through various changes throughout the years. In recent times, the existing models use K-Nearest Neighbors (KNN), ML algorithm Ref. [2, 25]. It is used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows. This algorithm has some disadvantages like

low accuracy, slow prediction, and are computationally expensive.

2.3 Detection of anomaly objects using CNN

Later, the researchers started adapting DL algorithms like Dual-Stage Detection, which has been the predominant approach and remains a powerful paradigm. The first stage generates the regions of interest, while the second stage classifies these proposals and locates the objects using bounding box regression. Examples include Convolutional Neural networks (CNN) Ref. [1, 7], Feature Pyramid Networks (FPN), Spatial Pyramid Pooling Networks (SPP-net), and the R-CNN Family (R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN) Ref. [3, ,18-20]. Some problems with this method is it requires a huge amount of data and it works well only with image data input.

2.4 Multi object detection using Single Stage Detectors

Single-Stage Detection is a paradigm that came into prominence with the advent of YOLO (You Only Look Once) Ref. [4, 9, 15, 21, 22], and subsequently gained significant traction with the Single Shot Multibox Detector (SSD) Ref. [5, 11, 17, 24], RetinaNet, and others. In this paradigm, the detector skips the region proposal stage, and directly classifies and locates the bounding boxes. This makes the single stage detection comparatively faster than dual stage detection Ref. [10, 16].

2.5 Drawbacks in existing models

Object detection and tracking is one of the critical areas of research. Hece, the issues that still persist are due to routine change in motion of objects, variation in scene size, occlusions and appearance variations. Real-time object detection with top-level classification and localization accuracy remains challenging. Change in viewpoint also makes it difficult to identify the object. Analysing the problems and challenges along with the existing solutions identified in this literature survey, we planned to deploy best suited algorithms into a mobile app to identify multiple objects at real time.

3. DATASETS

Object detection is inextricably linked to other similar computer vision techniques like image recognition and image segmentation and also helps us understand

and analyze scenes in images or video. But there are important differences. Image recognition only outputs a class label for an identified object, and image segmentation creates a pixel-level understanding of a scene's elements. What separates object detection from these other tasks is its unique ability to locate objects within an image or video.To perform these tasks effectively , highly precise and large datasets are needed. So, we use Google's open Images dataset , DUTS dataset and PASCAL-S dataset,COCO dataset.

3.1 Google's open Images dataset

This Open Images dataset is one of the largest existing datasets with object location annotations. Open Images is a dataset of around 9M images annotated with image-level labels, object bounding boxes, object segmentation masks, and visual relationships. It contains a total of 16M bounding boxes for 600 object classes on 1.9M images, making it the largest existing dataset with object location annotations.

3.2 COCO dataset

COCO is a large-scale dataset. COCO provides: Object segmentation, Recognition in context, Superpixel stuff segmentation, 330K images (>200K labeled), 1.5 million object instances, 80 categories.

3.3 DUTS dataset

DUTS is a saliency detection dataset containing 10,553 training images and 5,019 test images. All training images are collected from the ImageNet DET training/val sets, while test images are collected from the ImageNet DET test set and the SUN data set. Both the training and test set contain very challenging scenarios for saliency detection. Accurate pixel-level ground truths are manually annotated by 50 subjects.

3.4 PASCAL-S dataset

Pascal VOC provides standardized image data sets for object detection. PASCAL-S is a dataset for salient object detection consisting of a set of 850 images from PASCAL VOC 2010 validation set with multiple salient objects on the scenes. Unlike COCO dataset, Pascal-S is an XML file.

4. PROPOSED METHODOLOGY

Typically, there are three steps in an object detection framework.

1. First, a model or algorithm is used to generate regions of interest or region proposals. These region proposals are a large set of bounding boxes spanning the full image (that is, an object localisation component).
2. In the second step, visual features are extracted , they are evaluated and determine what objects are present in the proposals based on visual features (i.e. an object classification component).
3. In the final post-processing step, overlapping boxes are combined into a single bounding box (that is, non maximum suppression).

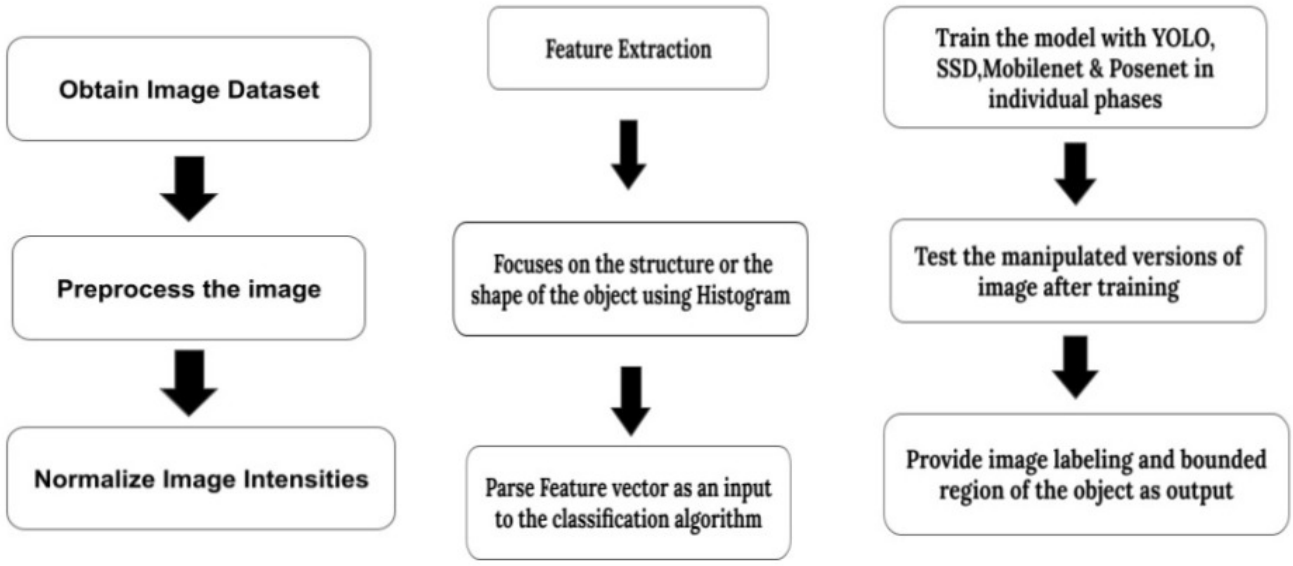

Fig 1. Phases involved in developing our research project

4.1 Feature extraction using Histogram of Gradients

The image is passed to the HOG or Histogram of Oriented Gradients, a feature detector to preprocess the image (adjust the image size, intensities, contrast in terms of illumination). It identifies each object uniquely into a set of features. The feature vector or feature map which is obtained after preprocessing is

passed as an input to the classification algorithm of Convolutional Neural Networks. The first step consists of dividing each image into blocks (grid). Then, for each detector window, we will be building gradient vectors using the properties of each pixel i.e. color and intensity.

4.2 Compute the Gradients from each pixel of the image

In this step, for a 2D surface, the gradient is derived by computing the derivative in terms of x and the derivative in terms of y. As much as this may sound complex, for an image, computing the gradient consists mainly of measuring, for each pixel, the variation of its surrounding pixels. “When taking the derivative of an image, we're actually taking what’s called a discrete derivative, and it’s more of an approximation of the derivative.” This variation is computed by applying a correlation mask in the x direction and another mask in the y direction.

4.3 Correlation filtering using Filters or masks to make modifications to image

Mask is a term used in correlation filtering. Filtering is the process of applying modifications to an image. In order to apply those changes, a filter/mask/kernel is used on each pixel. A mask is simply a matrix that contains the weights by which each pixel is affected by its surrounding pixels. Hence, the mask is used to re-compute the value of each pixel in a picture. The new value is the sum of the adjacent pixels with the mask weights taken into consideration. This type of filtering is the correlation filtering. Thus the preprocessed image is parsed to the Convolutional algorithms

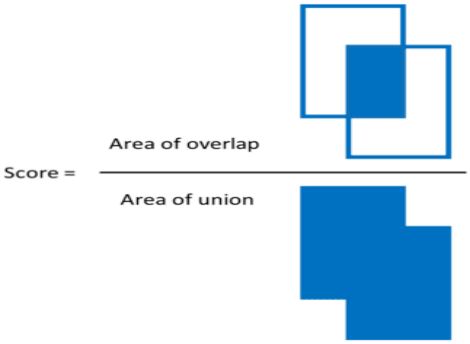

Fig 2. An example of selective search applied to an image. A threshold can be tuned in the SS algorithm to generate more or fewer proposals.

5. ARCHITECTURE DIAGRAM

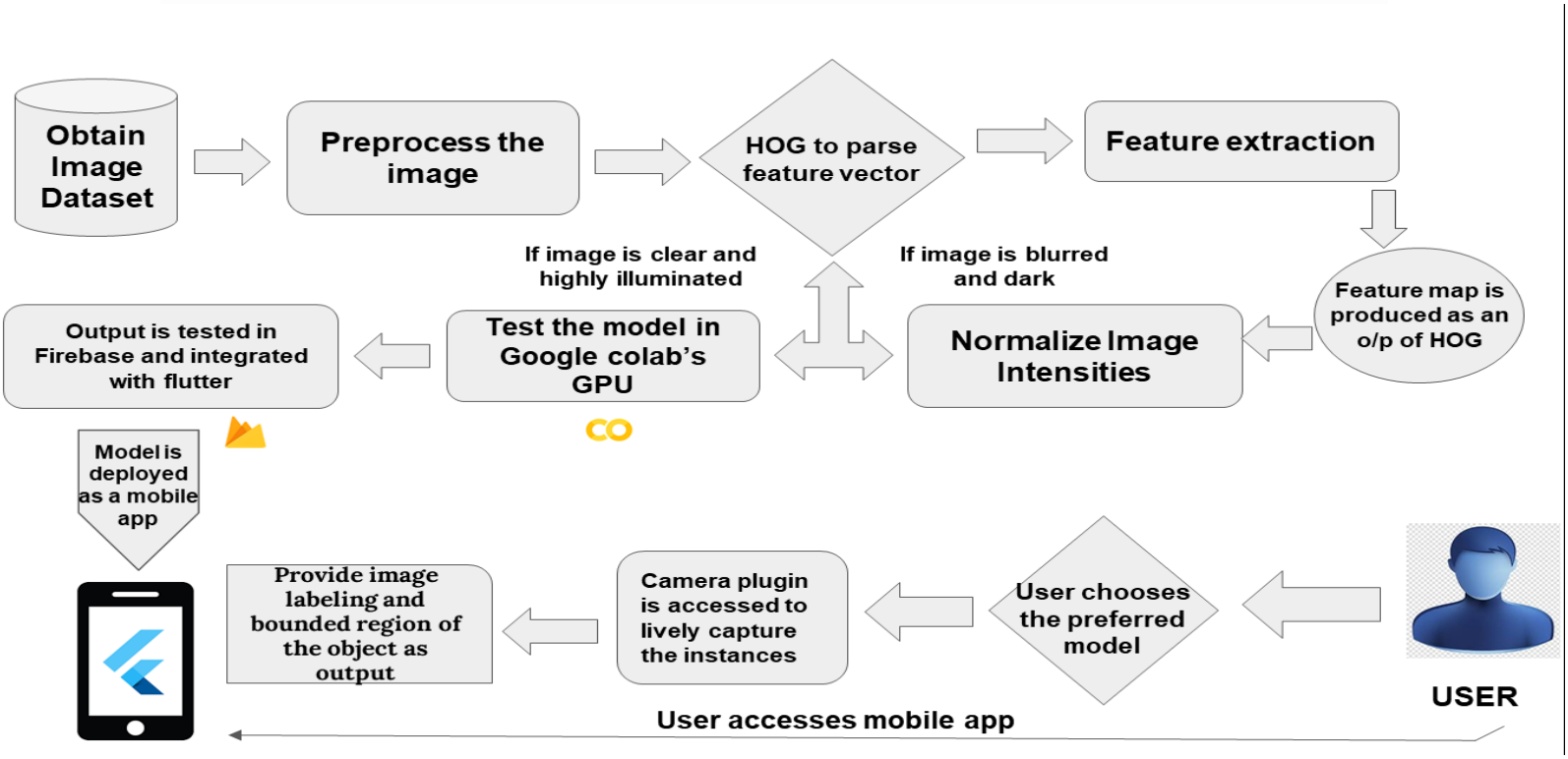

Fig 3. Network architecture diagram of object detection

6. ALGORITHMIC STEPS

6.1 Rich dataset training in YOLO

Drawing bounding boxes on images for object detection is much more expensive than tagging images for classification, the paper proposed a way to combine small object detection dataset with large ImageNet so that the model can be exposed to a much larger number of object categories. The name of YOLO9000 comes from the top 9000 classes in ImageNet. In order to efficiently merge ImageNet labels (1000 classes, fine-grained) with COCO/PASCAL (<100 classes, coarse-grained), YOLO9000 built a hierarchical tree structure with reference to WordNet so that general labels are closer to the root and the fine-grained class labels are leaves.

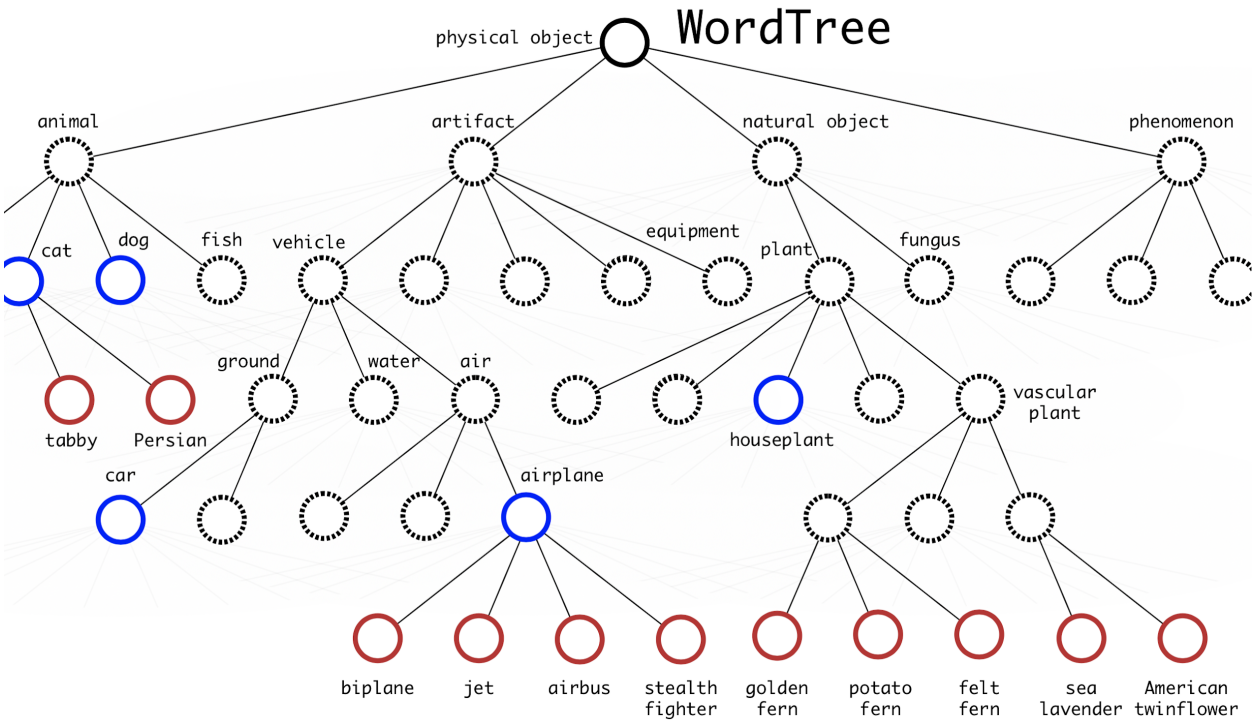

Fig 4. Hierarchical data training in YOLO

= Pr("persian cat" | contain a "physical object")

= Pr("persian cat" | "cat")

= Pr("cat" | "animal")

= Pr("animal" | "physical object")

= Pr(contain a "physical object")

6.2 SSD for multi-scale feature map prediction

At first,RCNN was used to detect objects from a single layer. Actually, it uses multiple layers (multi-scale feature maps) to detect objects independently. As CNN reduces the spatial dimension gradually, the resolution of the feature maps also decreases. SSD uses lower resolution layers to detect larger scale objects. For example, the 4× 4 feature maps are used for larger scale objects.The SSD object detection composes of 2 parts: Extracting feature maps, and Applying convolution filters to detect objects.

System	VOC2007 test *mAP*	FPS (Titan X)	Number of Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
YOLO (customized)	63.4	45	98	448 x 448
SSD300* (VGG16)	77.2	46	8732	300 x 300
SSD512* (VGG16)	**79.8**	19	24564	512 x 512

Table 1. Comparing accuracy of SSD and YOLO in terms of resolution.

6.3 Posenet and Mobilenet

Pose estimation is the task of using an ML model to estimate the pose of a person from an image or a video by estimating the spatial locations of key body joints (keypoints). The model estimates an X and Y coordinate for each keypoint. 3D pose estimation works to transform an object in a 2D image into a 3D object by adding a z-dimension to the prediction. 3D pose estimation allows us to predict the actual spatial positioning of a depicted person or object. Posenet is thus used for activity recognition, motion detection , Augmented reality and training robots. When posenet is used to track orientation of objects, mobilenet comes into existence.

Id	**Part**	**Id**	**Part**
1	nose	10	Left shoulder
2	leftEye	11	Right shoulder
3	rightEye	12	Left elbow
4	leftEar	13	Right elbow
5	rightEar	14	Left wrist
6	Right wrist	15	Left elbow
7	Right elbow	16	Left ankle
8	Right ankle	17	Hip region
9	Left knee	18	Right knee

Table 2 . Keypoints of human body

6.4 Tensorflow Detection API

The Tensorflow Detection API brings together a lot of the aforementioned ideas together in a single package, allowing you to quickly iterate over different configurations using the Tensorflow backend. With the API, you are defining the object detection model using configuration files, and the Tensorflow Detection API is responsible for structuring all the necessary elements together.

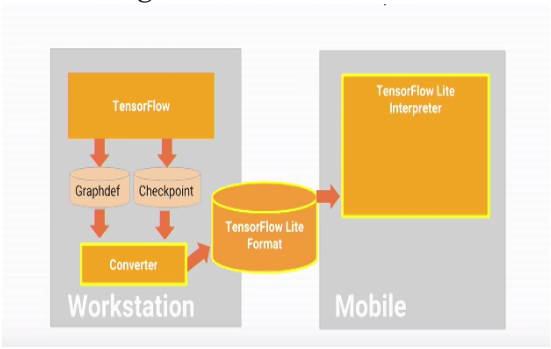

Fig 5 . Workflow of TensorFlow

6.5 Evaluating the performance of the model :

Use Precision and Recall as the metrics to evaluate the performance. Precision and Recall are calculated using true positives(TP), false positives(FP) and false negatives(FN)

6.6 Calculate the mean Average Precision(mAP), using 11 point interpolation technique:

Finally, we develop a flutter app with 4 modules, each having the functionality of an object detection model. We approach 4 models; they are SSD, YOLO, MobileNet and PoseNet. We use a camera plugin that captures live video, and the trained model identifies the objects visible on the screen. The output is displayed by labelling the image by bounding them within a box. Using the Stack widget, we can place the bounding boxes on top of the image. We can also display the detected class and its accuracy as percentage by simply adding a Text widget and converting the confidence to percentage.

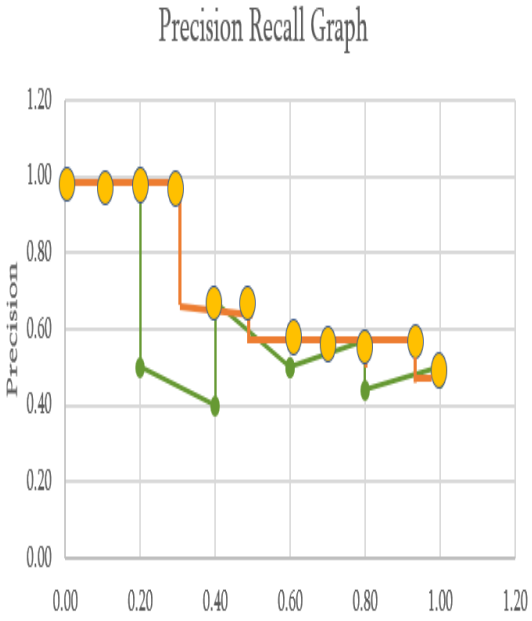

Fig 6 . Mean average precision graph

The mAP hence is the Mean of all the Average Precision values across all the classes as measured above. This is in essence how the Mean Average Precision is calculated for Object Detection evaluation. There might be some variation at times, for example the COCO evaluation is more strict, enforcing various metrics with various IOUs and object sizes.So, to conclude, mean average precision is, literally, the average of all the average precisions(APs) of our classes in the dataset.

7. RESULTS AND OBSERVATIONS

The proposed model is developed with the objective of accurately detecting the real time objects in live video. The model is trained using Tensorflow with an initial learning rate of 0.001, 0.9 momentum, 0.0005 weight decay, and batch size 32. Using an Nvidia Titan X on the VOC2007 test, SSD achieves 59 FPS with mAP 74.3% on the VOC2007 test, vs.Posenet 7 FPS with mAP 73.2% or YOLO 45 FPS with mAP 63.4%. The whole network is trained in an end-to-end fashion with the multi-task loss. We carry out several experiments on PASCAL and COCO datasets to demonstrate object detection with higher accuracy and efficiency. The resulting system is interactive and engaging. While the existing models processes images individually, but the deployed model functions like a tracking system, detecting objects as they move around .

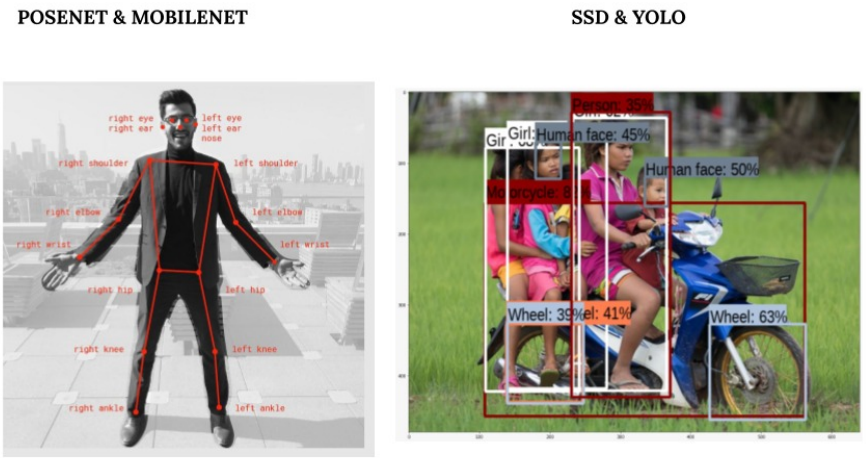

Fig 7 . Visualization of predicted models

8. CONCLUSION AND FUTURE WORKS

With the continuous upgrading of powerful computing equipment, object detection technology based on deep learning has been developed rapidly. In order to deploy on more accurate applications, the need for high precision real-time systems is becoming more and more urgent. Since achieving high accuracy and efficiency detectors is the ultimate goal, with the help of the effective object detection algorithms like SSD, YOLO, MobileNet and PoseNet the results of this project, achieves in providing a highly accurate detector that can detect objects quickly in real time. Our future work may include deploying the model to various embedded and IoT based devices to detect objects in a large frame.

REFERENCE

1. Object Detection Using Convolutional Neural Networks by Reagan L. Galvez, Argel A. Bandala, Elmer P. Dadios, Ryan Rhay P. Vicerra, Jose Martin Z. Maningo, 2018.
2. Object Detection of Surgical Instruments for Assistant Robot Surgeon using KNN by Fica Aida Nadhifatul Aini, Ahmad Zatnika Purwalaksana, Istas Pratomo Manalu, 2019.
3. Design and Implementation of an Object Detection System Using Faster R-CNN by Cheng Wang; Zhihao Peng, 2019.
4. YOLOrs: Object Detection in Multimodal Remote Sensing Imagery by Manish Sharma, Mayur Dhanaraj, Srivallabha Karnam, Dimitris G. Chachlakis, 2021.
5. SSD Object Detection Model Based on Multi-Frequency Feature Theory by Jinling Li; Qingshan Hou; Jinsheng Xing; Jianguo Ju, 2020.
6. Hierarchical Alternate Interaction Network for RGB-D Salient Object Detection by Gongyang Li; Zhi Liu; Minyu Chen, 2021.
7. Efficient Rail Area Detection Using Convolutional Neural Network by Zhangyu Wang; Xinkai Wu; Guizhen Yu, 2018.
8. Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors by Sasa Sambolek; Marina Ivasic-Kos, 2021.
9. YOLO-ACN: Focusing on Small Target and Occluded Object

Detection by Yongjun Li; Shasha Li; Haohao Du; Lijia Chen, 2020.

10. ReFPN-FCOS: One-Stage Object Detection for Feature Learning and Accurate Localization by Jiexian Zeng; Jiale Xiong; Xiang Fu, 2020.
11. SSD Performance Modeling Using Bottleneck Analysis by Jihun Kim; Joonsung Kim; Pyeongsu Park, 2019.
12. Occlusion Problem-Oriented Adversarial Faster-RCNN Scheme by Qingyang Xu; Xiaofeng Zhang; Ruoshi Cheng; Yong Song, 2019.
13. Fast Synthetic Dataset for Kitchen Object Segmentation in Deep Learning by Ruben Sagues-Tanco; Luis Benages-Pardo; Gonzalo López-Nicolás, 2020.
14. Driver Fatigue Detection Method Based on Eye States With Pupil and Iris Segmentation by Qianyang Zhuang; Zhang Kehua; Jiayi Wang; Qianqian Chen, 2020.
15. Pedestrian Detection Based on YOLO Network Model by Wenbo Lan; Jianwu Dang; Yangping Wang, 2018.
16. Multi-Target Tracking and Detection Based on Hybrid Filter Algorithm by Xianzhen Xu; Zhiyu Yuan; Yanping Wang, 2020.
17. SSD Object Detection Model Based on Multi-Frequency Feature Theory by Jinling Li; Qingshan Hou; Jinsheng Xing, 2018.
18. Research of Image Main Objects Detection Algorithm Based on Deep Learning by Liyan Yu; Xianqiao Chen; Sansan Zhou, 2018.
19. An IF-RCNN Algorithm Pedestrian Detection in Pedestrian Tunnels by Jin Ren; Changliu Niu; Jun Han, 2020.
20. Self-Enhanced R-CNNs for Human Detection With Semi-Supervised Assumptions by Xuexian Chen; Si Wu; Zhiwen Yu, 2020.
21. Automatic Detection of Melanoma with Yolo Deep Convolutional Neural Networks by Yali Nie; Paolo Sommella; Mattias O'Nils, 2019.
22. CPU Based YOLO: A Real Time Object Detection Algorithm by Md. Bahar Ullah, 2020.
23. Object Detection and Count of Objects in Image using Tensor Flow Object Detection API by B N Krishna Sai; T. Sasikala, 2019.
24. PoseNet Based Acupoint Recognition of Blind Massage Robot by Chen Chen; Ping Lu; Siqi Wang, 2020.
25. KNN-Based Approximate Outlier Detection Algorithm Over IoT Streaming Data by Rui Zhu; Xiaoling Ji; Danyang Yu; Zhiyuan Tan, 2020