Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 04, April 2021: 2336-2345

Research Article

Real Time Localized Multi Object Detection System

T. Sethukarasi[1]., K C. Sakthi Siva Parvathi[2], S. Supraja Arthi[3], R. Yuvasree[4]

[1,2,3,4]Department of Computer Science and Engineering, RMK Engineering College, Chennai 601206, India

Abstract

Object discovery is a key skill required in many computer and robotic viewing programs. Recent research in this area has made great strides in many areas. Our project focuses on building a flutter-based mobile app with a beautiful UI using some popular object detection algorithms such as SSD, YOLO, MobileNet and PoseNet. We use real-time data available from Google's open image data sets, COCO, DUTS and PASCAL. To mark the performance of our project, we are training our models with Google Colab's GPU and TensorFlow acquisition API. The trained models are then converted into lightweight TensorFlow Lite files and included in our project guide. We install a camera plugin that allows the camera on our mobile to capture ongoing events and based on the selected algorithm, marking a bounding box around objects and recording it with high accuracy. In this way, our research project makes it easier for ordinary people to use it on a daily basis and discover hidden or suspicious objects in the environment.

Keywords: *Object Recognition,TensorFlow, Labels, Accuracy*

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 project is to identify the predefined sets of object classes like people, cars, fruits, utensils and define the location of each item found in the image using the matching box. Flutter integrated with Firebase facilitates 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 and section 8.

Literature Survey

Detailed research was conducted under the "Computer vision" domain. In this study, we now know that most of them focus on finding and counting items at the scene Ref. [6], determine and follow human behavior Ref. [8, 14, 19] and their exact locations, all while labeling them accurately.

2.1 Challenges in smaller objects detection

Object detection is considered more difficult than image classification,because of the great challenges that remain. Although there are many barriers, researchers have found creative solutions and have put forth great effort to overcome these difficulties, amazing results are not fully realized. Object detection frameworks continue to fight with small things, especially those that are close and have partial occlusions Ref. [12, 13]. There are a few challenges such as multiple scale factoring, speed of the object, limited data, and class imbalance that make object detection impossible.

2.2 Object detection using KNN

Object detection models have undergone various changes over the years. In more recent times, existing models use K-Nearest Neighbors (KNN), ML algorithm Ref. [2, 25]. Both classification and regression based problems were solved using the KNN algorithm. It is easy to use and understand, but there is a big problem with huge delays as the size of the data used grows. This algorithm has some problems such as low accuracy, slow prediction, and 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 R-CNN Family Ref. [3, 18-20]. Some problems with this method require a large amount of data and only work well with image capture.

2.4 Multiple acquisition 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], and subsequently gained significant traction with the Single Shot Multibox Detector (SSD) Ref. [5, 11, 17], 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 discovery and tracking is one of the most critical areas for research. Hence, the issues that still persist are due the usual change in the movement of objects, variations in the size of the screen, variations and differences in its appearance. 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.

Datasets

Object detection differs from these other tasks like image recognition and image classification with its unique ability to locate objects within an image or video. This is possible because of the datasets with which object detection models are trained. To detect objects, the model needs to understand and analyze the scenes happening. In order 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 Open Images Dataset

This open image database is one of the largest widgets available with object annotations. Open images refers to a dataset of images that contains around 9M defined by image-level labels and visual relationships. It contains a total of 16M bounding boxes for 600 categories of objects in 1.9M images, making it the largest database available with object annotations.

3.2 COCO Dataset

COCO or in other words Common Objects in Context, is another big dataset. COCO offers asset classification, contextual recognition, super pixel object classification and 330,000 or 330K images, 15,00,000 or 1.5 million object shapes, 80 categories.

3.3 DUTS dataset

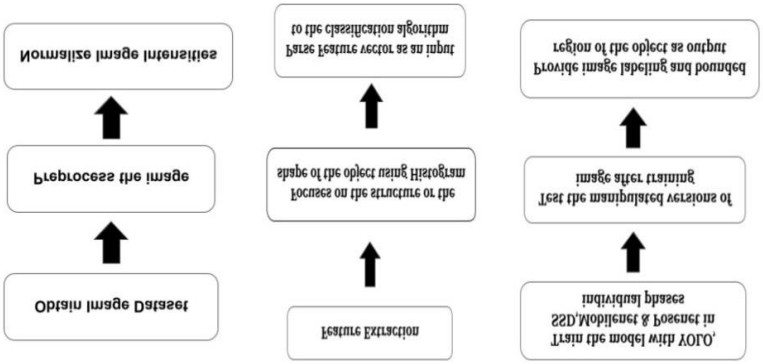
DUTS is a salient detection dataset that approximately contains 10,000 training images and 5,000 test images. All training images are collected in ImageNet DET training sets, while test images are collected in the ImageNet DET test set and SUN data set. Both the training and

testing dataset contain the most challenging conditions for salient object detection. Accurate pixel-level facts are presented in 50 subjects.

3.4 PASCAL-S dataset

Pascal VOC provides the same image data sets for object detection. PASCAL-S is a dataset of 850 images for salient object detection. It is from PASCAL VOC 2010 validation set with many of the salient elements in the scene. Unlike the COCO database, Pascal-S is an XML file.

Proposed Methodology



Generally, there are three main steps in the object detection framework. At first, a model or algorithm is used to generate regions of interest or regional proposals. These are a large set of bounding boxes that include a full image eg. a homemade object. In the second step, visual element features are extracted, evaluated and determined. In the final step of processing, the interlocking boxes are grouped into a single bounding box for continuous pressure.

Figure 1. Categories involved in the implementation of 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 segmentation algorithm. The first step is to separate each image into blocks (grid) and then in each detector window, we will create gradient vectors using the properties of each pixel i.e. color and durability.

4.2 Compute the gradients from each pixel of an image

In this step, on the 2D surface, the gradient is calculated by computing the derivative in terms of x and based on y. Although this may sound complicated computing the gradient consists of measuring for each pixel, the variation of its surrounding pixels. “When we take a picture-

based approach, we actually take what is called a copyright, and it's more than almost anything else." These variations are calculated by using a composite mask in the x direction and another mask in direction y.

4.3 Correlation filtering using filters or masks to perform image modification

A mask is a word used in filtering. Filtering is the process of modifying an image. To apply those changes, a filter/ kernel is applied to each pixel. It is just a matrix containing the weights per pixel that are affected by the surrounding pixels. Therefore, it is used to recalculate the number of pixels per image. The new value is the sum of the nearest pixels with the estimated weight of the mask. The previously created image is then transferred to Convolutional algorithms

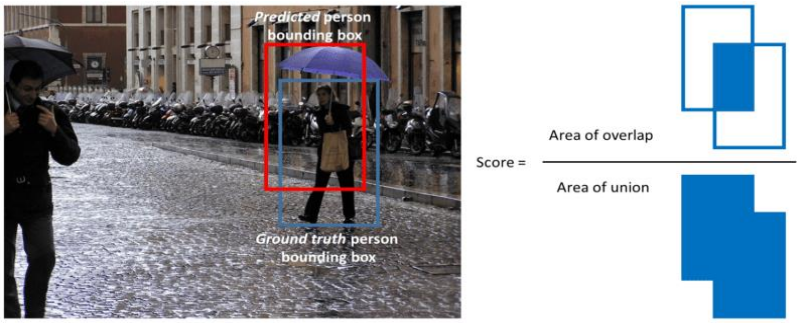


Figure 2. Example of a selected search used in an image.

The limit can be adjusted to the SS algorithm to produce more or less suggestions

Architecture Diagram

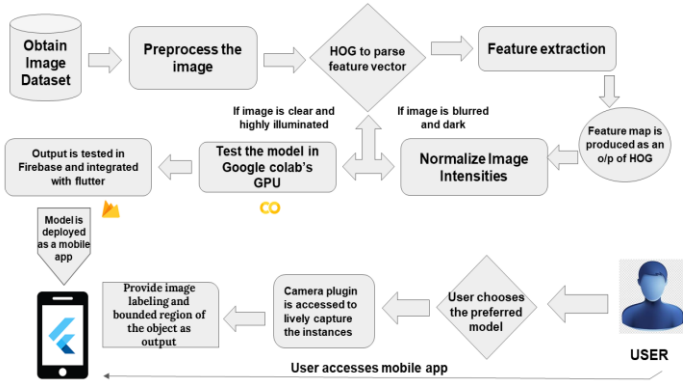


Figure 3 Diagram of the network construction of an acquisition object

Algorithmic Steps

6.1 Training the datasets in YOLO

Drawing bounding boxes on the detected images are more expensive than marking separate images. The paper suggests a way to combine a small acquisition dataset with a larger ImageNet so that the model is presented to a larger number of object categories. The name YOLO9000 appears in the top 9000 classes in ImageNet. To better integrate ImageNet labels with COCO / PASCAL (<100 classes, long leaves), YOLO9000 builds a tree-based hierarchical structure based on WordNet so that standard labels are close to root and labels

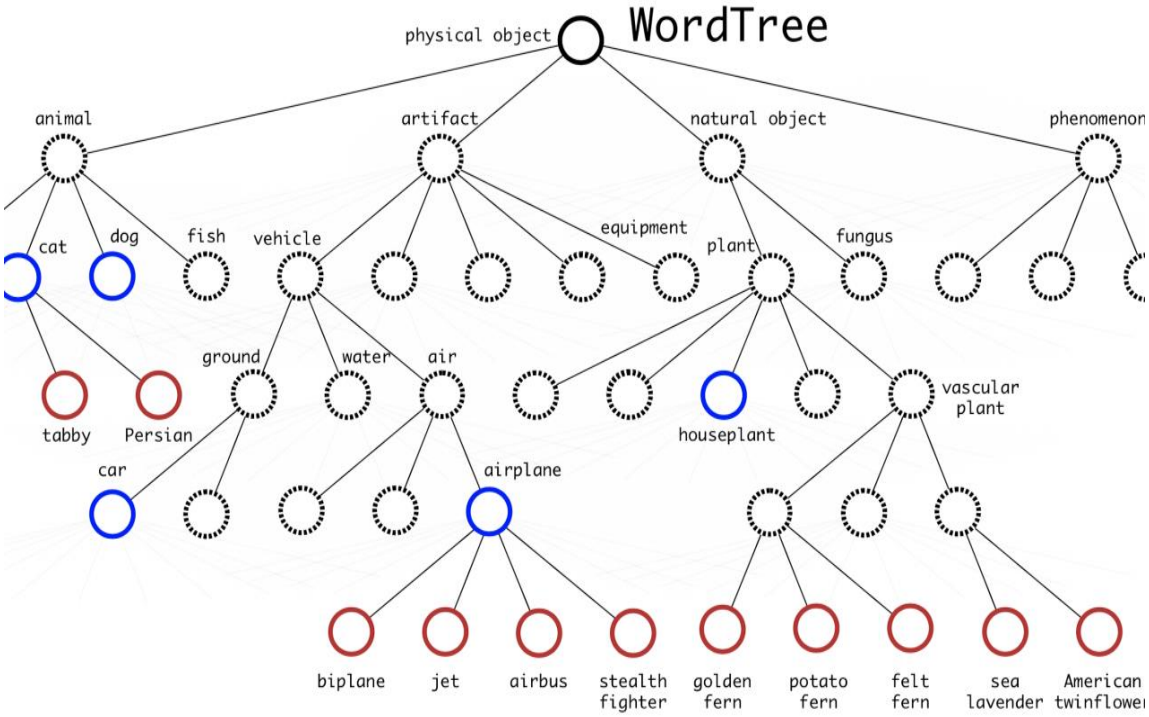


Figure 4 Data training downloaded from YOLO

= Pr ("Persian cat" | contains "object")

= Pr ("Persian cat" | "cat")

= Pr ("cat" | "animal")

= Pr ("animal" | "object")

6.2 SSD for multi-scale feature map prediction

Initially, RCNN was used to obtain items from a single layer. In fact, it uses many layers to locate objects independently. As CNN gradually reduced the size of spatial dimension, resolution also decreased. SSD uses a feature map matrix for large scale objects. The SSD model comprises 2 parts: Extracting feature maps, and Applying convolution filters to detect objects.

Table 1: Comparison of SSD and YOLO accuracy with resolution

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

6.3 Posenet and Mobilenet

Pose measurement is the function of using the ML model to measure a person's posture from a photo or video by measuring areas with important body joints. The model measures the X and Y coordinates of each keypoint. 3D pose estimation is used to convert a 2D image into a 3D object by adding the size of the z 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.

Table 2: Key Points of human body

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

6.4 Tensorflow Detection API

The Tensorflow Detection API brings together many of the above concepts in a single package, allowing you to swiftly iterate over different configurations using the Tensorflow backend. With the API, we define the object acquisition model using configuration files and the TensorFlow Object Detection API is responsible for building all the required items together. 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.

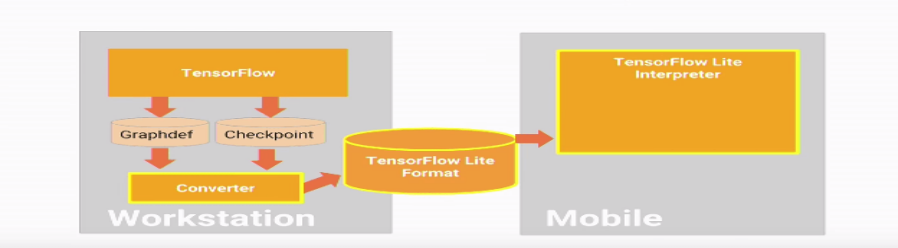


Figure 5 TensorFlow workflow

6.5 Evaluating the performance of the model

We use Precision and Recall as performance testing metrics to identify the performance of the models. Precision and Recall are calculated using true positives (TP), false positives (FP) and negative negatives (FN)

6.6 Calculate the mean Average Precision(mAP)

The mAP, mean Average Precision id evaluated for calculating the performance of object detection. It has values across all these classes. Evaluation of COCO dataset is more difficult since it uses various metrics for objects of variable size. Thus mAP is the average of all the average precisions(APs) of our classes in the dataset.

Results and Observations

The proposed model is developed for the purpose of accurately detecting real-time objects in live video. The model is trained using TensorFlow with a first reading level of 0.001 to 0.9 intensity and 0.0005 weight loss. After testing and training the modals, SSD achieves around 74% mAP, Posenet achieves with mAP 72%, YOLO achieves mAP with 63% and lastly MobilenNet achieves 69% mAP. The whole network is trained end to end with a minimal loss of multiple jobs. We did a lot of research using various datasets to show the detection of

an object with high accuracy and efficiency. The emerging system is interactive and attractive. Thus, the project we implemented acts as a tracking system, for detecting objects that are both stationary and in movement.

Figure 7 Visualization of the predicted models

Conclusion and Future Works

With the continuous development of powerful computer technology, object recognition technology based on in-depth learning is rapidly developing. In order to use more accurate applications, the need for more accurate and more accurate applications is becoming more and more urgent. As the acquisition of high accuracy and efficiency equipment is a major goal, with the help of effective acquisition systems such as SSD, YOLO, MobileNet and PoseNet the results of this project, it is possible to provide the most accurate detector that can get things faster in real time. Our future work may include installing a model on various embedded devices with IoT to access objects in a large frame

References

[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] FPN-FCOS One-Stage Object Detection that is used for Feature Learning and Localization by Jiale Xiong; Xiang Fu;Jiexian Zeng;, 2020.
[11] Single Stage Detection Performance Modeling Using Bottleneck Analysis by Jihun Kim; Pyeongsu Park; Joonsung Kim;, 2019.
[12] Occlusion Problem-Oriented Faster R-Convolutional Neural Network Scheme by Qingyang Xu; Ruoshi Cheng; Yong Song; Xiaofeng Zhang; , 2019.
[13] Fast Synthetic Dataset for the Kitchen Object Segmentation using DL by Luis Benages-Pardo; Ruben Sagues-Tanco; Gonzalo López-Nicolás; 2020.
[14] Driver Fatigue Detection Method based on Eye With Pupil and Iris Segmentation by Zhang Kehua; Qianqian Chen Jiayi Wang; Qianyang Zhuang, 2020.
[15] Pedestrian Detection using YOLO Network Model by Yangping Wang, Wenbo Lan; Jianwu Dang; 2018.
[16] Multi-Target Tracking and Detection on Hybrid Filter Algorithm by Xianzhen Xu; Yanping Wang; Zhiyu Yuan;, 2020.
[17] Single Stage Object Detection Model based on Multi-Frequency Feature Theory byQingshan Hou; Jinsheng Xing; Jinling Li, 2018.
[18] Research of Image Main Objects Detection Algorithm Based on Deep Learning by Xianqiao Chen; Liyan Yu; Sansan, 2018.
[19] An IF-R CNN Algorithm Pedestrian Detection in Pedestrian Tunnels by Jin Ren; Changliu Niu; 2020.
[20] Self-Enhanced R-CNNs that is used for Human Detection which uses Semi-Supervised Assumptions by Zhiwen Yu Si Wu;Xuexian Chen, 2020.