REAL TIME OBJECT DETECTION

A PROJECT REPORT

Submitted by

Manas Shukla (20MIM10003)
Sanskar Tale (20MIM10018)
Rajnish Mishra (20MIM10097)
Pankaj Sunil Patil (20MIM10113)

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRI KALAN, SEHORE
MADHYA PRADESH - 466114

DEC 2021

BONAFIDE CERTIFICATE

Certified that this project report titled **“Real-Time Object Detection''** is the bonafide work of “**Manas Shukla (20MIM10003), Sanskar Tale (20MIM10018), Rajnish Mishra (20MIM10097) and Pankaj Sunil Patil (20MIM10113)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr Mayuri AVR, Senior Assistant Professor
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Dr J. Manikandan, Assistant Professor
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr V. Pandimurugan, Head of the Department, School of Computing Science and Engineering for much of his valuable support and encouragement in carrying out this work.

I would like to thank my internal guide Dr J. Manikandan, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

YOLO	You Only Look Once
mAP	Mean Average Precision
BoF	Bag of Freebies
BoS	Bag of Supplies
URL	Uniform Resource Locator
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
RPN	Region Proposal Network
AI	Artificial Intelligence
ML	Machine Learning
SVM	Support Vector Machine
GPU	Graphics Processing Unit
VPU	Vision Processing Unit
IDE	Integrated Development Environment
OpenCV	Open Source Computer Vision Library
CSP	Cross-Stage-Partial
ReLU	Rectified Linear Unit

LIST OF FIGURES AND GRAPHS

ABSTRACT

Object detection is one of the primary tasks in computer vision which consists of determining the location on the image where certain objects are present, as well as classifying those objects. In 2015, the YOLO (You Only Look Once) algorithm was born with a new approach, reframing object detection as a regression problem and performing in a single neural network. That made the object detection field explode and obtained much more remarkable achievements than just a decade ago. So far, combined with many of the most innovative ideas coming out of the computer vision research community, YOLO has been upgraded to five versions and assessed as one of the outstanding object detection algorithms. The 5th generation of YOLO, YOLOv5, is the latest version not developed by the original author of YOLO. However, the performance of the YOLOv5 is higher than the YOLOv4 in terms of both accuracy and speed.

TABLE OF CONTENTS

CHAPTER - 1

Project Description and Outline

1.1 Introduction

When people look at an image, they can immediately recognise what the things are and where they are in the image. The capacity to identify objects quickly mixed with a person's knowledge aids in making an appropriate judgement on the object's nature. Scientists are working on a system that can imitate the ability of the human visual system to detect items. The two criteria for evaluating an object detection algorithm are speed and accuracy.

One of the most well-known challenges in computer vision is object detection. It not only classifies but also identifies the object in the image. The methods used to solve this problem in prior decades consisted of two stages: (1) extracting distinct regions of the image using sliding windows of various widths, and (2) applying the classification problem to identify what class the objects belong to. These methods have the drawback of requiring a lot of processing and being split down into several phases. As a result, speed optimization of the system is challenging.

1.2 YOLO – YOU ONLY LOOK ONCE

With research published in 2015 by Joseph Redmon et al., YOLO entered the computer vision landscape. "You Only Look Once: Unified, Real-Time Object Detection" drew a lot of interest from other computer vision experts right away. Prior to the invention of YOLO, Convolutional Neural Networks (CNN) such as Region Convolutional Network (R-CNN) used Regions Proposal Networks (RPNs) to produce proposal bounding boxes on the input image, then run a classifier on the bounding boxes, and then apply post-processing to remove duplicate detections and refine the bounding boxes. Individual levels of the R-CNN network could not be trained independently. It was tough and time-consuming to optimize the R-CNN network.

1.3 Problem Statement

Object detection is the problem of finding and classifying a variable number of objects on an image. The important difference is the “variable” part. In contrast with problems like classification, the output of object detection is variable in length, since the number of objects detected may change from image to image.

1.4 Objective

The project “Object Detection System using Machine Learning Technique” detects objects efficiently based on the YOLO algorithm and applies the algorithm on image data and video data to detect objects.

CHAPTER - 2

Related Work Investigation

2.1 Introduction

The field of object detection is not as new as it may seem. Object detection has evolved over the past 20 years. The progress of object detection is usually separated into two separate historical periods. The CNN framework is an important model for deep learning theory, with a wide range of applications in image recognition and classification. It is developed from artificial neural networks. The previous layer is used as the input of the subsequent layer, and the back-propagation algorithm is used to update the parameters. The CNN model contains many network layers, can take the original image as the input, and may subsequently introduce many practical strategies, such as convolution, pooling and dropout, to Real-Time Improve the fault tolerance of the model. Among these, convolution and pooling are necessary strategies in existing CNN models.

2.2 Core Area of the Project

The computer vision system is a core area of this project During the last years, there has been a rapid and successful expansion of computer vision research. Parts of this success have come from adopting and adapting machine learning methods, while others from the development of new representations and models for specific computer vision problems or from the development of efficient solutions. One area that has attained great progress is object detection. The present works give a perspective on object detection research.

2.3 Existing Approach

Faster RCNN

Faster R-CNN is an object detection model that improves on Fast R-CNN by utilising a region proposal network (RPN) with the CNN model. The RPN shares full-image convolutional features with the detection network, enabling nearly cost-free region proposals. It is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. RPN and Fast R-CNN are merged into a single network by sharing their convolutional features: the RPN component tells the unified network where to look.

As a whole, Faster R-CNN consists of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector that uses the proposed regions.

2.4 (i) Pros of Faster RCNN

- Reduced the total number of initial features for CNN, from 6,000,000 to only 3000.
- Instead of 2000 SVMs, we are classifying using Softmax functions of quantities equivalent to the number of classes. Softmax generally performs better than SVMs.

2.4 (ii) Cons of Faster RCNN

- It uses the Selective Search Algorithm to find the Regions of Interest which is a slow and time-consuming process.
- It takes around 2 seconds per image to detect objects, which sometimes does not work properly with large real-life datasets.

Chapter - 3

3.1 Introduction

To run the object detection model, we need good software for coding and better hardware for the smooth functioning of the model. The coding software helps the model to run efficiently and rectify any errors that occur while running the code and the hardware ensures that the model runs with an efficient speed as well as maintain accuracy of the model.

3.2 Hardware and Software Requirements

Hardware Requirements

Processor - Intel i7 9700k

Clock speed - 4.90 GHz

GPU - Nvidia GTX 1080ti/ RTX2080

RAM - 8GB

Software requirements

Distribution - Anaconda Navigator

Library - TensorFlow, OpenCV, YOLO

IDE - Jupyter Notebook, Google - Colab

Language - Python

GPU Architecture - CUDA

3.3 Specific Project requirements

3.3.1 Data Requirement

We require a large amount of data for this project. For the training and validation, we will be using images of 8 classes - 'Raccoon'; 'Bucket'; 'TableFan'; 'Sunglasses'; 'Mask'; 'Dog'; 'Sneakers' and 'Orangutan'. The images for this dataset are downloaded from websites like Roboflow and Kaggle which provide a good number of images for training purposes and the videos are downloaded from youtube for testing the model.

Chapter - 4

Design Methodology

4.1 Methodology

The basic aim is the fast operating speed of the neural networks, in production systems and optimization for parallel computations, rather than the low computation volume theoretical indicator (BFLOP). We present two options of real-time neural networks:

• For GPU we use a small number of groups (1 - 8) in convolutional layers: CSPResNeXt50 / CSPDarknet53

• For VPU - we use grouped-convolution, but we refrain from using Squeeze-and-excitement (SE) blocks - specifically this includes the following models: EfficientNet-lite / MixNet [76] / GhostNet [21] / MobileNetV3

For improving the object detection training, a CNN usually uses the following:

• Activations: ReLU, leaky-ReLU, parametric-ReLU, ReLU6, SELU, Swish, or Mish

• Bounding box regression loss: MSE, IoU, GIoU, CIoU, DIoU • Data augmentation: CutOut, MixUp, CutMix

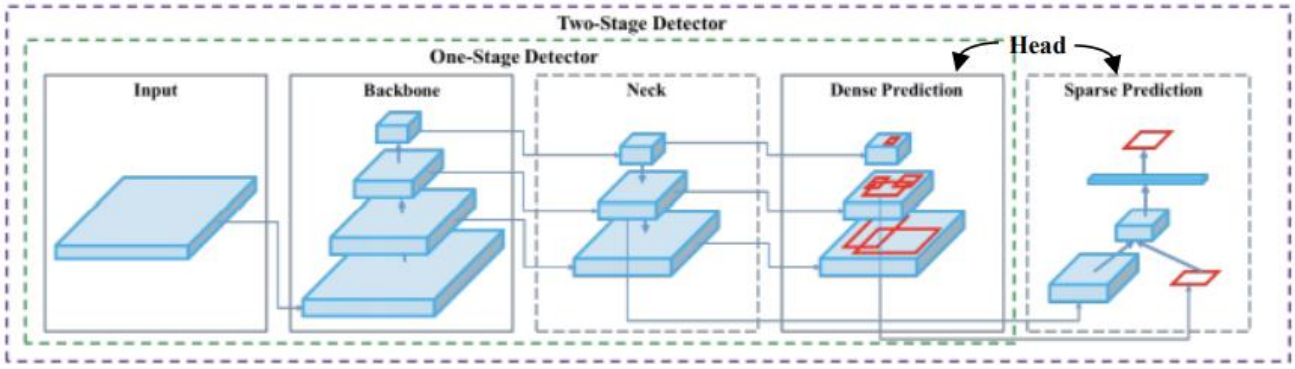
• Regularization method: DropOut, DropPath [36], Spatial DropOut [79], or DropBlock

• Normalization of the network activations by their mean and variance: Batch Normalization (BN) [32], Cross-GPU Batch Normalization (CGBN or SyncBN) [93], Filter Response Normalization (FRN) [70], or Cross-Iteration Batch Normalization (CBN) [89]

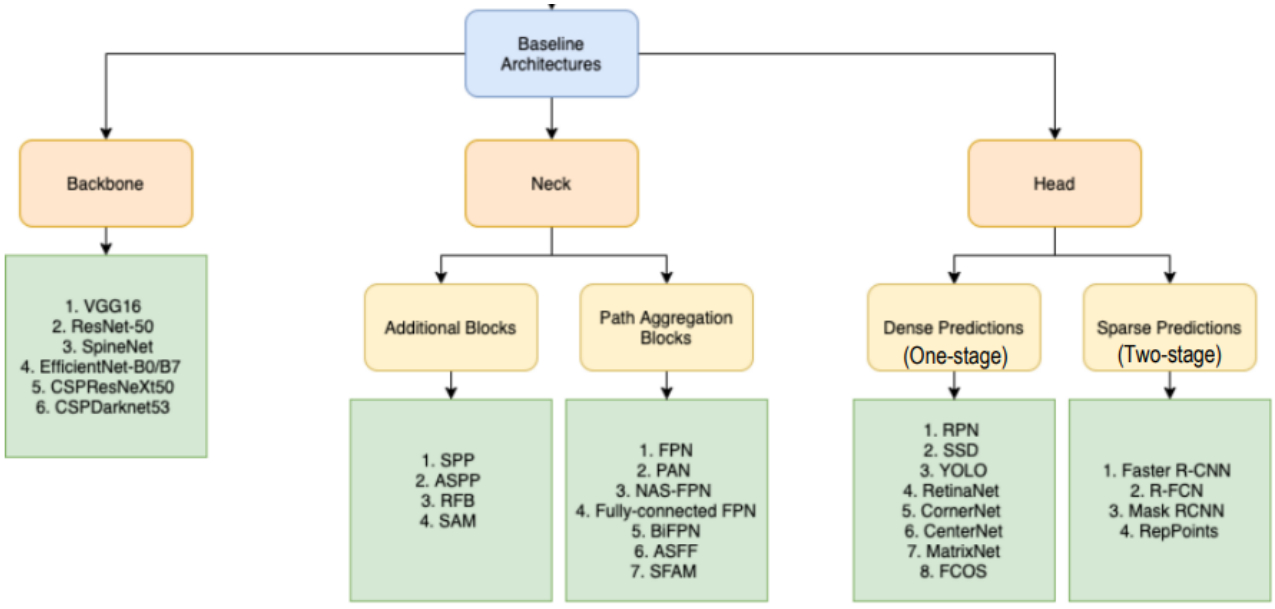
• Skip-connections: Residual connections, Weighted residual connections, Multi-input weighted residual connections, or Cross stage partial connections (CSP)

4.2 Software Architecture Designs

The common point of all object detection architectures is that the input image features will be compressed down through the feature extractor (Backbone) and then forwarded to the object detector (including Detection Neck and Detection Head) as in Figure 15. Detection Neck (or Neck) works as a feature aggregation which is tasked to mix and combine the features formed in the Backbone to prepare for the detection step in Detection Head (or Head)



(Fig. 1)



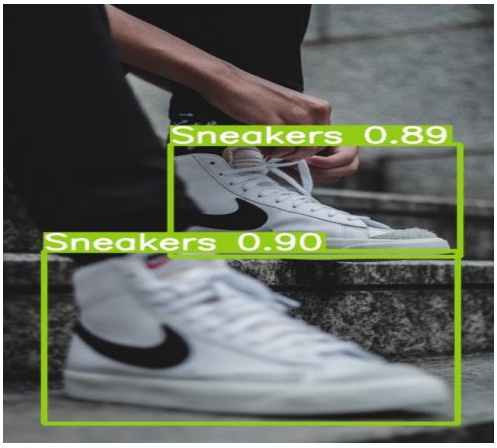
(Fig. 2)

Chapter - 5

Technical Implementation and Analysis

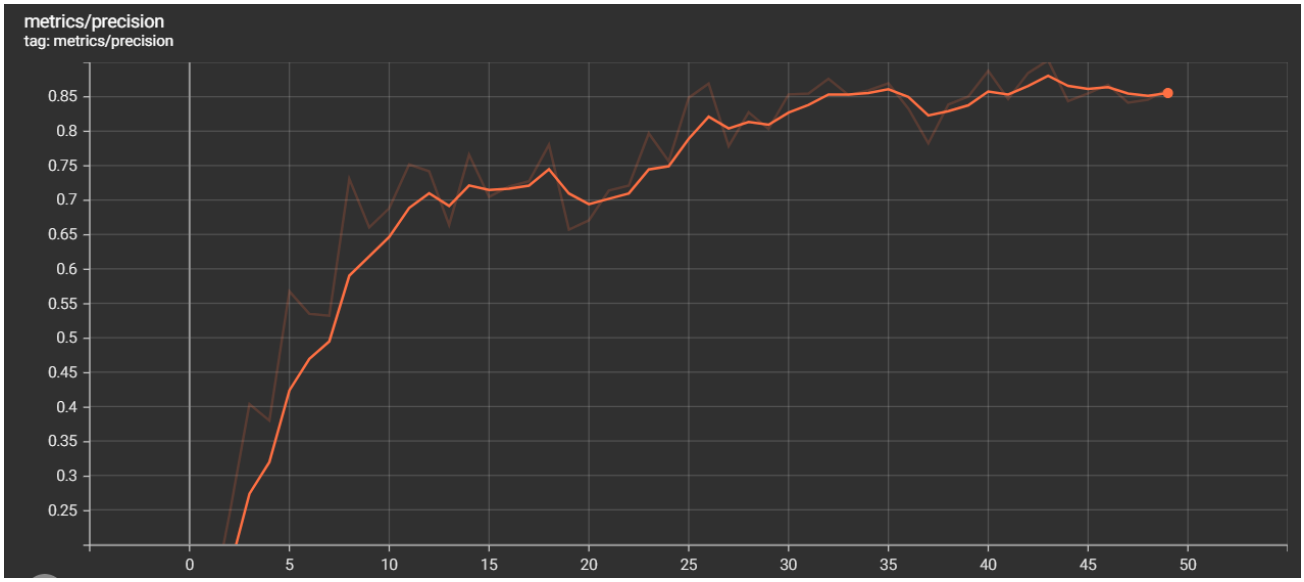
5.1 Test and Validation

Some of the outputs received after running the model are as shown below:



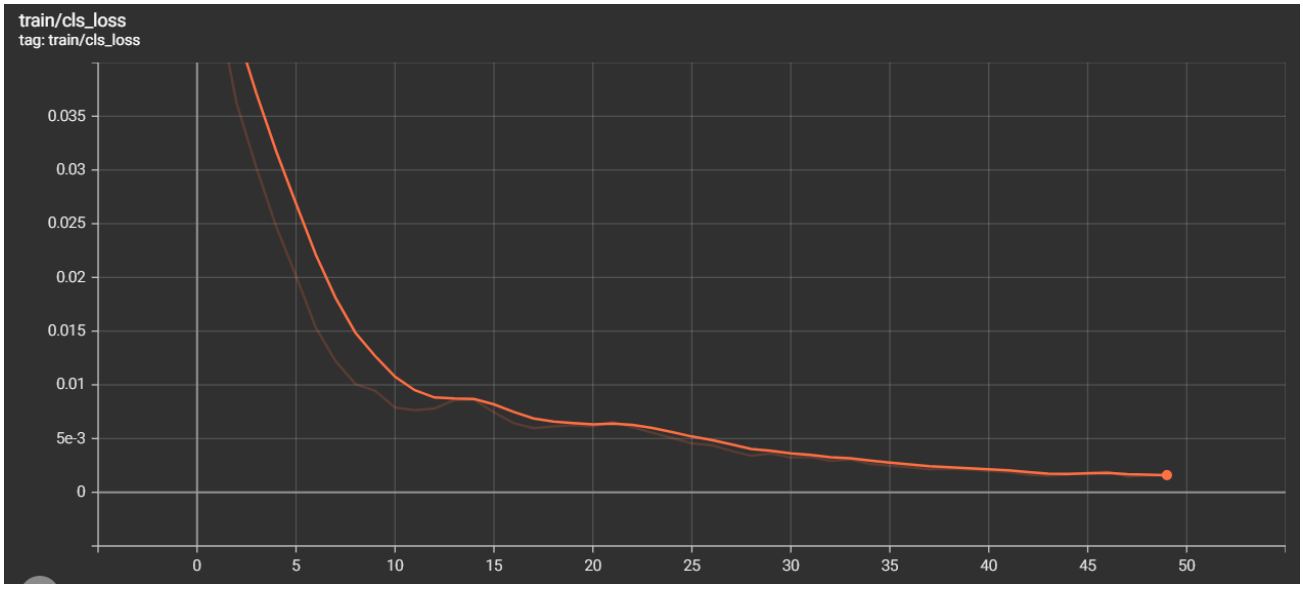
5.2 Performance Analysis

mAP (Mean Average Precision)



(Fig. 3)

Class Loss



(Fig. 4)

Chapter - 6

Project Outcome and Applicability

6.1 Key Implementation Outline

We can distinguish between these three computer vision tasks with this example:
Image Classification: This is done by predicting the type or class of an object in an image.
Input: An image that consists of a single object, such as a photograph.
Output: A class label (e.g. one or more integers that are mapped to class labels).Object Localization: This is done by locating the presence of objects in an image and indicating their location with a bounding box.
Input: An image that consists of one or more objects, such as a photograph.
Output: One or more bounding boxes (e.g. defined by a point, width, and height).Object Detection: This is done through, Locate the presence of objects with a bounding box and types or classes of the located objects in an image
Input: An image that consists of one or more objects, such as a photograph.Output: One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box.

6.2 Significant Project Outcome

The project's purpose is to ensure that the operations of a few sectors and professions run smoothly without causing financial harm. Because manpower is required and workplace safety must be assured under severe pandemic COVID-19 settings, such models are necessary and efficient to implement. Apart from that, for the sake of safety and sanity, public transit, such as airports, can be equipped with similar systems.
With the help of this model we can ensure the safety of the people in public places. As people who are not wearing masks can get sick and can transmit it to others also. Mask enforcement is necessary for public safety and using "Real-Time Object Detection" is one easy method.

6.3 Project applicability on Real-world applications

1. Video Surveillance

With object tracking, it would be easier to track a person in a video. Object tracking could also be used in tracking the motion of a ball during a match. In the field of traffic monitoring to object tracking plays a crucial role.

2. Anomaly Detection

In the field of agriculture, object detection helps in identifying infected crops and thereby helps the farmers take measures accordingly. It could also help identify skin problems in healthcare.

3. Self Driving Cars

For a car to decide what to do in the next step, whether to accelerate, apply brakes or turn, it needs to know where all the objects are around the car and what those objects are. That requires object detection and we would essentially train the car to detect a known set of objects such as cars, pedestrians, traffic lights, road signs, bicycles, motorcycles, etc.

4. Object Recognition as Image Search

By Recognizing the objects in the images, combining each object in the image and passing detected objects labelled in the URL we can make the object detection system an image search.

5. Robotics

Autonomous assistive robots must be provided with the ability to process visual data in real-time so that they can react adequately to quickly adapt to changes in the environment. Reliable object detection and recognition is usually a necessary early step to achieve this goal.

Chapter - 7

Conclusion and Recommendation

7.1 Outline

Nowadays, there is still a lot of controversy about the name and improvements of YOLOv5 in the computer vision community about innovations that have not made a breakthrough. However, the name aside, the performance of YOLOv5 is at least not inferior to the YOLOv4 in both speed and accuracy. With the built-in Pytorch framework that is user-friendly and has a larger community than the Darknet framework, there is no doubt that YOLOv5 will receive more contributions and have more growth potential in the future.

The field of computer vision, especially object detection, has only exploded in the last 5 years or so. Therefore, although it has evolved over 5 generations and is one of the outstanding object detection algorithms, the YOLO algorithm is still flawed. Therefore, an AI system cannot be built from a mere algorithm, it is necessary to integrate more optimization methods and the most state-of-the-art ideas in the field of computer vision to help the AI system achieve the best performance.

7.2 Limitation/Constraints of the System: -

⇒ Viewpoint Variation

An object viewed from different angles may look completely different. This is one of the challenges with object detection because most detectors are trained with images only from a particular viewpoint.

⇒ Deformation

Many objects of interest are not rigid bodies and can be deformed in extreme ways. If an object is deformed extremely, the object detector might not be able to detect and identify it.

⇒ Occlusion

The objects of interest can be occluded. Sometimes only a small portion of an object, as little as a few pixels could be visible. For example, when a person is holding a mobile phone, it is a challenge to detect it in this situation.

⇒ Illumination Conditions

The effects of illumination are drastic on the pixel level. Objects exhibit different colours under different illumination conditions. For example, an outdoor surveillance camera is exposed to different lighting conditions throughout the day, including bright daylight, evening, and night light. This affects the capability of the detector to detect objects robustly.

⇒ Cluttered or Textured Background

The objects of interest may sometimes blend into the background, making them hard to identify. For example, the cat and dog in these images are camouflaged with the rug they are sitting/lying on. In these cases, object detectors will face challenges detecting cats and dogs.

7.3 Future Enhancement

Object detection is a key task for most computer and robot vision systems. Although there has been great progress in the last several years, there will be even bigger improvements in the future with the advent of artificial intelligence in conjunction with existing techniques that are now part of many consumer electronics or have been integrated into short-term assistant driving technologies.

However, we are still far from achieving human-level performance in open-world learning.

Furthermore, object detection has not been applied in many areas where it could be of great help. Consider for example the possibility of applications of object detection systems to robotic excavation when venturing into previously unexplored territory, such as the deep sea or other planets, in which the detection systems will have to learn new object classes on the job. In such cases, a real-time, open-world learning ability will be critical.

This fascinating computer technology related to computer vision and image processing that detects and defines objects, such as persons, vehicles, and animals from digital images and videos, will be incredibly important in the near future.

REFERENCES

1. Glenn Jocher, Official Yolov5 repository: https://github.com/ultralytics/yolov5
2. Do Thuan, Evolution of Yolo Algorithm and Yolov5: https://www.theseus.fi/bitstream/handle/10024/452552/Do_Thuan.pdf?sequence=2&isAllowed=y
3. Priya Dwivedi, Comparison of Yolov5 and Faster RCNN: https://towardsdatascience.com/yolov5-compared-to-faster-rcnn-who-wins-a771cd6c9fb4
4. Alexander Fleiss, The future of Object Detection: https://blog.rebellionresearch.com/blog/the-future-of-object-detection
5. Sabina Pokhrel, Limitations of Object Detection: https://towardsdatascience.com/6-obstacles-to-robust-object-detection-6802140302ef
6. Roboflow, To download large number of images for the custom dataset: https://public.roboflow.com/
7. MakeSense, To annotate and label the images for the dataset: https://www.makesense.ai/