

DETECTION AND ANALYSIS OF OBJECTS IN VIDEO SEQUENCES

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

KOLLA OM VIVEK (Reg.No – 39110511)
KOLLA OM VITESH (Reg.No- 39110510)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with 'A' grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KOLLA OM VIVEK (Reg.No - 39110511)** and **KOLLA OM VITESH (Reg.No – 39110510)** who carried out the Project Phase-2 entitled “**DETECTION AND ANALYSIS OF OBJECTS IN VIDEO SEQUENCES**” under my supervision from January 2023 to April 2023.

Internal Guide

Dr. D. USHA NANDINI M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.



Submitted for Viva voce Examination held on 20.04.2023

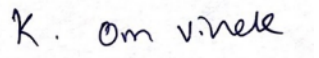
Internal Examiner

External Examiner

DECLARATION

I, **KOLLA OM VIVEK(39110511)**, hereby declare that the Project Phase-2 Report entitled "**DETECTION AND ANALYSIS OF OBJECTS IN VIDEO SEQUENCES**" done by me under the guidance of **Dr. D. Usha Nandini, M.E.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 20.04.2023
PLACE: Chennai


SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr.L.Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. D. Usha Nandini M.E., Ph.D**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 projectwork.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTARCT

Identifying and analyzing elements in the video can be difficult due to changes in lighting, the appearance of the subject, and comparable off-target elements in the background. In this study, we used YOLOv5 to extract and classify objects in the video. A dataset called "Detection" was created using Python approaches for detecting objects. The detection dataset contains many categories of "x" photographs. The YOLOv5 models were adjusted and refined to identify real-time items such as humans, dogs, rifles, etc. An annotated Detection dataset was used to train YOLOv5, which has been fine-tuned for faster performance and improved detection accuracy. The created model was used to determine bounding boxes for the objects in the video. Additionally, each object is identified by its name and highlighted with a different color, making object recognition easier. Object detection is widely used in computer vision and is crucial for various applications, e.g., self-driving cars, military, security surveillance etc. During the development of half a century, object detection methods have been continuously developed and generated numerous approaches that have obtained promising results. At present, the object detection approach has largely evolved into two categories: traditional machine learning methods utilizing varied computer vision techniques and deep learning methods. In this thesis, we put forward several contributions to deal with the problems of detecting and tracking multi-objects in video sequences. The proposed frameworks are based on deep learning networks and transfer learning approaches. The objective of object tracking is to associate target objects in consecutive video frames. Object tracking requires the location and shape or features of objects in the video frames. So, object detection and object classification are the preceding steps of object tracking in computer vision applications. To detect or locate the moving object in the frame, object detection is the first stage in tracking. After that, the detected objects can be classified as vehicles, humans, swaying trees, birds, and other moving objects. It is a challenging task in image processing to track objects into consecutive frames. Complex object motion, irregular object shape, occlusion of object to object and object to the scene, and real-time processing requirements can all pose problems. Object tracking has a variety of uses, some of which are: surveillance and security, traffic monitoring, video communication, robot vision, and animation.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLES	ix
	LIST OF ABBREVIATIONS	x
1	INTRODUCTION	1
2	LITERATURE SURVEY	3
	2.1 Inferences from Literature Survey	4
	2.1.1 yolov5	4
	2.2 Open problems in Existing System	5
	2.2.1 Viewpoint Variation	5
	2.2.2 Deformation	6
	2.2.3 Occlusion	6
	2.2.4 Textured background	7
	2.2.5 Speed	7
3	REQUIREMENTS ANALYSIS	8
	3.1 Feasibility Studies of the Project	8
	3.1.1 Detecting small objects	8
	3.1.2 Speed	8
	3.1.3 Accuracy of the object	8
	3.1.4 Test Time Augmentation	8
	3.2 Requirements Specification	9
	3.2.1 Hardware Requirements	9
	3.2.2 Software Requirements	9
4	DESCRIPTION OF EXISTING SYSTEM	10
	4.1 The cons are listed below	12
	4.1.1 Illumination challenges	12
	4.1.2 Speed	12
	4.1.3 Unpredicted motion	12
	4.1.4 Occlusion	13

	4.2 Working of YOU ONLY LOOK ONCE (YOLO)	14
	4.2.1 Hyper-parameters used	15
	4.2.2 CNN architecture of Darknet-53	15
	4.2.3 Layers Details	16
	4.2.4 Convolution layers in YOLOv5	16
5	DESCRIPTION OF PROPOSED WORK	17
	5.1 Dataset	17
	5.2 Process Model	18
	5.2.1 Object detection	18
	5.2.2 Focus layer	20
	5.2.3 Convolution Layers	20
	5.2.4 BottleneckCSP Module	22
	5.2.4.1 The effect of BottleneckCSP	22
	5.2.5 SPP (spatial pyramid pooling layer)	23
	5.2.5.1 YOLO with SPP	23
	5.3 Architecture	24
	5.4 Advantages	27
	5.5 Analysis	27
	5.6 Project Management Plan	27
	5.7 Financial report on estimated costing	27
6	RESULTS AND DISCUSSION	28
7	CONCLUSION	30
	7.1 Future Work	31
	7.2 Research Issues	32
	7.3 Implementation Issues	33
	REFERENCES	35
	APPENDIX	37
	A. SOURCE CODE	37
	B. SCREENSHOTS	63
	C. RESEARCH PAPER	68

FIGURE NO.	FIGURE NAME	PAGE NO.
2.1	Yolov5	4
2.2	Actual image	5
2.3	DetectedObjects	5
2.4	Viewpoint	6
2.5	Deformation	6
2.6	Occlusion	7
2.7	Textured background	7
4.1	Yolo	14
4.2	Prediction map	15
5.1	Sample for labelling images	17
5.2	Standard focus layer of YOLO-v5 model.	20
5.3	Convolution layer of YOLO-v5 model	21
5.4	Spatial pyramid pooling layer	23
5.5	YOLO with SPP	24
5.6	Flow diagram	24
5.7	Modified Yolov5 Architecture	25
5.8	The identity will go through different weight layers with relu	25
6.1	Data set	28
6.2	Detected objects	29

LIST OF FIGURES

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
1	Summarization about the existing system, methodology, and results	11
2	surface layers value for broken, hot_spot, black_border, scratch, and no_electricity	23
3	Summarization about the Epoch with respect to loss functions and mAP50	26

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	EXPANSION
1	YOLO	You Only Look Once
2	CNN	convolutional neural networks
3	SVM	support vector machines

CHAPTER 1

INTRODUCTION

In many applications, including visual surveillance, PPE detection, anomaly and defect identification, traffic monitoring and road maintenance, human-computer interaction, ADA compliance, and others, object recognition and analysis in video frames are now becoming a very important area.

The primary purpose of object detection is to recognize and estimate the class of an object from a given picture in a video sequence. Computer vision enables computers and software to obtain digital information about an image or video. Encoding objects play a vital role in computer vision technologies. Therefore, detection accuracy with the basal false positive rate is essential.

However, real-time detection in video sequences is technically difficult and time-consuming to do on a series of images due to several factors. One of them is that the images are made up of a large amount of labeled data that must be processed through complex operations. Since the advent of the video coding standard, analysis has become particularly important in allowing content-based image coding in this field.

The advent of learning technology has changed centuries of object detection and identification methods, providing comprehensive information about detected objects. Owing to the variety of uses, object detection has recently gained prominence. For example, in the field of anomaly and defect detection, the ability to automatically identify and classify anomalies or defects can help improve quality assurance consistency as well as the efficiency and effectiveness of production assembly.

All data about flaws and abnormalities is retained in the system by the detecting tools. The machine can draw inferences from it and improve its detecting capabilities over time. Whereas in the case of the classic/typical defect and anomaly detection

system, the efficacy of quality perception might plummet with each staff shift – resulting in expenditures.

For example, in the realm of human activities, People Counting- Counting people automatically have plural uses in smart cities and public spaces. Tracking the overall number of attendees to events or public attractions is crucial for safety and planning. In transportation systems, object detection may be used to determine the most popular bus or subway stations and offer statistics on route and capacity adjustments.

The capacity to precisely extract characteristics and patterns from the visual input is essential for object recognition to be successful. Object stability, or the capacity to recognize an element under various viewing circumstances, is a crucial component of object recognition. Object orientation, lighting, and object diversity (size, color, and other differences in the catalog) are a few of these many situations. The visual system must be able to infer the similarity of an object description across different retinal views and descriptions to achieve object consistency.

Edge detection and texture analysis are two examples of low-level image processing methods that are frequently used in conjunction with advanced machine learning algorithms like support vector machines (SVM) and You Only Look Once version 5 (YOLOv5). It is feasible to attain high levels of accuracy and resilience in real-world circumstances by training these algorithms on huge datasets of annotated activity samples.

CHAPTER 2

LITERATURE SURVEY

Object detection and analysis in video sequences has been the subject of numerous studies in recent years. Research on object detection is compiled in this section. Rosli and others proposed a YOLOv5 model by optimizing the hyper parameter for underwater detection [1]. They trained the Yolov5 model with a dataset containing various bright and blur images to check the performance of the algorithm based on quantitative results and frame rate. Based on momentum and learning rate, the feature extraction phase's hyper parameter was adjusted and further enhanced by ADAM in yolov5. The model's accuracy increased by 98.6%, and its frame rate increased by 106 frames per second as a result. ADAM has a learning rate and momentum of 0.0001 and 0.99, respectively, achieving greater accuracy for detecting objects underwater. Shan Luo and Jihong Liu [2] proposed an improved YOLOv5m and LPRNet model for car licence plate recognition. They used K-means++ and the DIOU loss function to improve YOLOV5M while removing the 20 x 20 feature map. After that, they went to LPRNet, which used to recognize characters on the licence plate. We got the model by combining the improved Yolov5m and LPRNet. The accuracy improved to 99.47%. By adding three more components to the existing YOLOV4 model, Chenglong Wang presented an improved YOLOV4 model for metal surface defect identification [3]. The first component among the three is self-dependent attentive fusion (SAF), which helps improve inter-path and feature fusion, followed by component-randomized mosaic augmentation, which helps find the over-transformed image, and the last one is perturbation-agnostic, which helps with regularization. Based on the proposed model, we found that YOLOv4 had 6.51% and YOLOv5 had 3.76% after validating it. Lingren Kong proposed a novel-based model called Yolo-g, it is a compact model for enhancing military object detection performance [4]. When we're discussing militaries, it's all about target tracking and analyzing the battlefield situation. It's already known that with the help of deep learning techniques, it's not possible to create a good model for military targets. So, Lingren Kong took the already existing YOLOv3 and added GhostNet, a compact CNN to the extracting features network. Which helps in the military's ability to detect targets quickly and accurately. With the help of the DIOU loss function, he

redesigned the loss function for target detection. Comparison to the original YOLOv3 algorithm, experimental findings demonstrate that our technique increases the detection rate by 25.9 images per sec and the MAP by 2.9%, respectively and the suggested model's size is reduced by 1/6 of YOLOv3's.

2.1 INFERENCES FROM LITREATURE SURVEY

2.1.1 YOLOv5

YOLOv5 (You Only Look Once, Version 5) is a real-time object detection algorithm that identifies specific objects in videos, live feeds, or images.

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.

YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects. This means that prediction in the entire image is done in a single algorithm run.

The biggest advantage of YOLOv5 in arcgis. Learn is that it comes preloaded with weights pretrained on the COCO dataset. This makes it ready-to-use for the 80 common objects (car, truck, person, etc.) that are part of the COCO dataset.

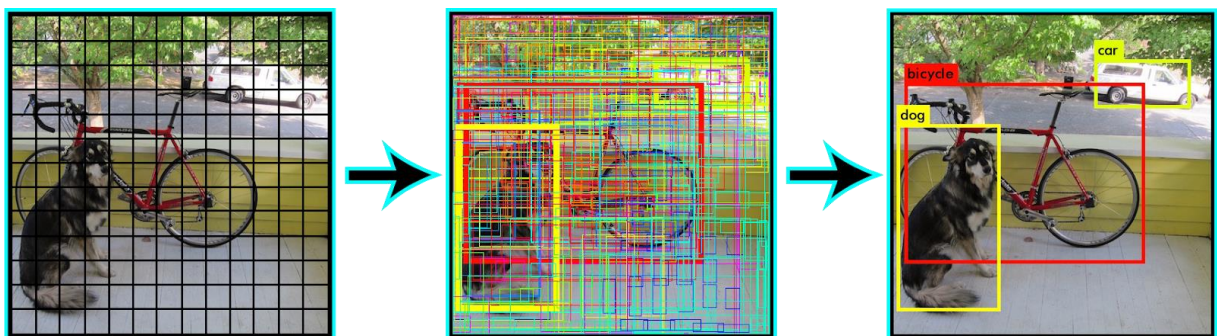


Fig. 2.1: Yolov5

Fig.2.1 shows how the yolov5 splits the input image into grids and tries to detect the object in all the grids which enables to detect the entire object

Actual Image:



Fig. 2.2: Actual image

Fig.2.2 represents the plain image where the yolov5 has to detect the objects

Image after Detection:

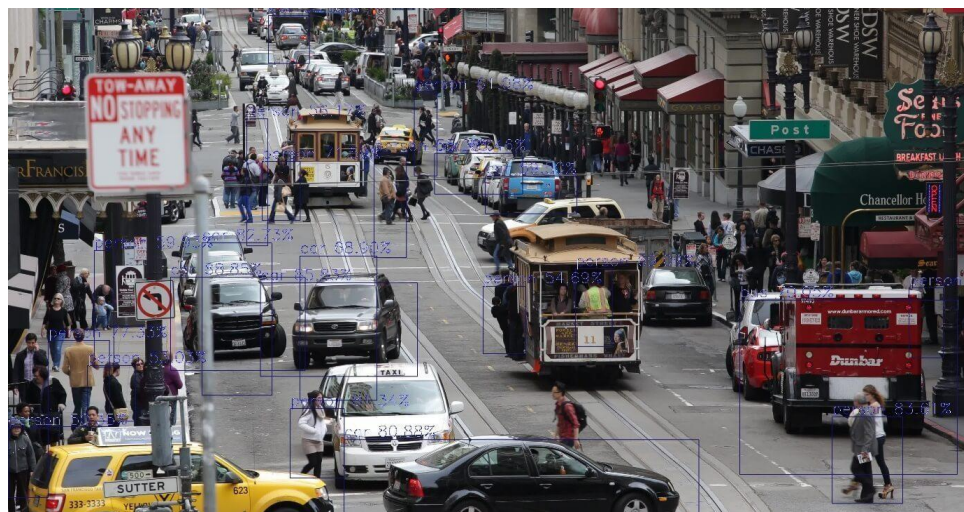


Fig. 2.3: Detected objects

Fig.2.3 represents the objects detected image where the yolov5 has detected the objects

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

2.2.1 Viewpoint Variation

One of the biggest difficulties of object detection is that an object viewed from different angles may look completely different. For example, the images of the cakes that you can see below differ from each other because they show the object from different sides.

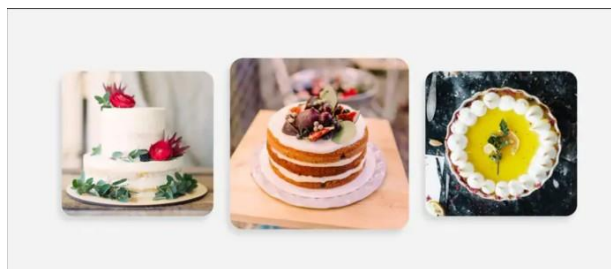


Fig. 2.4: Viewpoint

Fig.2.4 represents how we can view the cake from different sides

2.2.2 Deformation

The subject of computer vision analysis is not only a solid object but also bodies that can be deformed and change their shapes, which provides additional complexity for object detection.



Fig. 2.5: Deformation

Fig.2.5 represents the football players in different poses. If the object detector is trained to find a person only in a standing or running position, it may not be able to detect a player who is lying on the field or preparing to make a maneuver by bending down.

2.2.3 Occlusion

Sometimes objects can be obscured by other things, which makes it difficult to read the signs and identify these objects. For example, in the Fig.2.6 first below image, a cup is covered by the hand of the person holding this cup.



Fig. 2.6: Occlusion

In the Fig.2.6 second image, a person is also holding a mobile phone in such a way that the hands are occluding the object. Such situations create additional difficulties for determining the subject.

2.2.4 Cluttered or textured background

Objects that need to be identified may blend into the background, making it difficult to identify them. For example, the below Fig.2.7 picture shows a lot of items, the location of which is confusing when identifying scissors or other items of interest. In such cases, the object detector will encounter detection problems.



Fig. 2.7: Textured background

2.2.5 Speed

When it comes to video, detectors need to be trained to perform analysis in an ever-changing environment. It means that object detection algorithms must not only

accurately classify important objects but also be incredibly fast during prediction to be able to identify objects that are in motion.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 FEASIBILITY STUDIES OF THE PROJECT

3.1.1 Detecting Small Objects

Detecting small objects is one of the most challenging and important problems in computer vision.

3.1.2 Speed

When it comes to video, detectors need to be trained to perform analysis in an ever-changing environment. It means that object detection algorithms must not only accurately classify important objects but also be incredibly fast during prediction to be able to identify objects that are in motion

3.1.3 Accuracy of the object

The Compute Accuracy For Object Detection tool calculates the accuracy of a deep learning model by comparing the detected objects from the Detect Objects Using Deep Learning tool to ground reference data.

3.1.4 Test Time Augmentation

This is a process of sending augmented variations of a test image several times to the model and average the predictions of each image and return the final prediction instead of sending a clean image once and return the prediction as final. This will really help boost the accuracy of the model. Existing models apply Softmax technique

to compute a probability distribution for classes. But there's a risk of the model becoming too confident in its predictions which can result to over-fitting.

3.2 REQUIREMENTS SPECIFICATION

3.2.1 Hardware Requirements

- System : Intel® Core™ i5-9300H CPU @ 2.40GHz.
- Monitor : LED.
- Mouse : Logitech.
- Ram : 8.00 GB or above 8.00 GB
- Hard Disk : 1 TB

3.2.2 Software Requirements

- Operating System : Windows 10
- Language : Python 3
- Chrome Extension : Image Downloader
- Web Application : Jupyter Notebook
- Cloud based frame work : Google colab
- Cloud based storage : Google Drive

CHAPTER 4

DESCRIPTION OF EXISTING SYSTEM

Many Detecting objects in video sequence methods emerged from image-based detectors since video is made up of frames of pictures. Detecting and analyzing objects in video has the virtue of not being altered by environmental factors, and it is a prime spot for research in detection of objects in video sequence. The main works of detecting and analyzing objects in a video sequence are as follows:

- (1) An accomplishment of the spatiotemporal circumstances that form the setting for an information of broadcasting of moving visual images to strengthen accuracy;
- (2) tackle unusual and difficult video problems using local or global a vital piece of information (e.g., occlusion, unusual posture, etc.) and
- (3) To boost performance, decrease the processing of superfluous sections between video frames.

There are two types of Detectors that we can use, they are described as follows:

Image-Based: anchor-based (e.g. YOLOv5) and anchor-free are the two classified types of present state-of-the-art [SOTA] or significant detectors. The majority of them are built on a solid capability of extracting picture information. They have a diverse set of applications and settings. When used directly for video detection, these detectors tend to perform badly. In lieu, they don't take use of videos distinctive &

vital content. On the flip side, they lack specific enhancement for low-quality frames. To a wide range of detectors, post-processing approach is applicable. While employing video spatiotemporal context to increase detector accuracy. It lowers the rate of missed detection, increases the rate of erroneous detection, and enhances resilience.

Video-Based: Several technological approaches to video object detection have recently been developed. Following the flow-based concept, DFF and THP are proposed by X Zhu. Convolution networks and optical combination is the approach used; precisely, in order to avoid costly computations for non-important frames, the model conducts sophisticated intricacy and feature aggregation on a few critical sets (or portions in the frame) and computation of flow of optics on parts. It should be noted that the author of THP [10] recommends a strategy based on learning for distinguishing important areas from non-critical zones, This prompted us to develop a learning-based method for determining the similarity of thoughts.

They are broadly classified into two types: box and feature are the two slevel methods. They use temporal information on the box level in the box level approaches. The anticipated bounding boxes each frame is connected in a sequence along the time axis. The original score dispersion is then used to re-score these linked boxes.

Table 4.1: Summarization of the existing system, methodology, and results

Author	Data	Algorithm	Result
Iza Sazanits Isa	Underwater Detection	Yolov5	90%
Jihong Liu	License Plate Recognition	YOLOv5m	87%
Chenglong Wang	Metal Surface Defect	YOLOv4	82%
Jianzhong Wang	Military Targets	YOLO-G	84%
Ziying Song	Fine-Grained Birds	YOLOv3	79%

Li-Qun Zhou	Aerial Image	YOLOv3	84%
Lianbing Deng	ship detection	YOLOv3	90%
Jingyi Zhao	Vision-Based Vehicle	YOLOv4	83%
Lizong Liu	Transmission Lines	YOLOv3	87%
Zhuang-Zhuang Wang	Small-Object	YOLO and Dense	83%
Zhonghua Hong	ship detection	YOLOv3	90%
Chengyuan Song	Strawberry Fruit Growth	YOLO	85%
Mansheng Xiao	Freight Train	BD-YOLO	90%
Jyun-Yu Jhang	Traffic-Monitoring	YOLO	88%
Xiaorong Xu	Infrared Image	YOLOv5	83%

Object detection has an additional purpose in addition to classifying picture objects; this additional duty is known as object localization and is the first significant complexity of object detection. Researchers most frequently employ a multi-task loss function to solve this problem by penalizing both classification and localization failures.

$$L(p, u, t^u, v) = L_c(p, u) + \lambda [u \geq 1] L_l(t^u, v) \quad \square(1)$$

The equation represents the classification (L_c) and localization (L_l).

It is possible to change the λ to give categorization or localization a higher priority.

4.1 THE CONS ARE LISTED BELOW

4.1.1 Illumination challenges

Rapid changes in lighting could result in the detection of false positive objects. For instance, indoor lighting may suddenly turn on or off, or the light source itself may shift. The shadows cast by moving objects, reflections off of shiny objects, and rapid transitions from bright sunlight to gloomy or rainy conditions are all possible outside. Also, there is always a chance that a moving object and the background will share the same hue. To prevent errors in the detection of moving objects, the backdrop model should be able to adjust to variations in illumination and rapid changes in

brightness.

4.1.2 Speed

Detectors must be trained to do analysis in a constantly changing environment when it comes to video. This means that to be able to recognize items that are moving, object detection algorithms must not only properly categorize relevant objects but also be extremely quick during prediction.

4.1.3 Unpredicted motion

It is difficult to recognize items moving suddenly. In particular, a vehicle's jackrabbit start might cause a tracker to lose track of the item or result in a tracking algorithm mistake. Objects that move either too slowly or too quickly are another source of detecting problems. The temporal differencing approach won't be able to identify an object's parts if it travels slowly. As an item moves quickly, a trail of ghost regions will appear after it in the foreground mask. Another difficulty is intermittent motion, which is when an item moves for a time, pauses and then resumes moving.

4.1.4 Occlusion

Moreover, occlusions can make it far more challenging to find and follow moving objects in videos. For example in the case, a car driving along the road could get buried by some tree branches or other things. Object tracking techniques are made more difficult by the possibility of objects in a video feed being completely or partially obscured.

You Only Look Once or YOLO is one of the popular algorithms in object detection used by researchers around the globe.

It is based on the idea that :

“A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. “.

According to the researchers at Facebook AI Research, the unified architecture of YOLO is extremely fast in manner. The base YOLO model processes images in real-time at 45 frames per second, while the smaller version of the network, Fast YOLO processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. This algorithm outperforms the other detection methods, including DPM and R-CNN.

4.2 Working of YOU ONLY LOOK ONCE (YOLO)

The inputs is a batch of images of shape (m, 416, 416, 3).

YOLO v5 passes this image to a convolutional neural network (CNN).

The last two dimensions of the above output are flattened to get an output volume of (19, 19, 425):

- Here, each cell of a 19 x 19 grid returns 425 numbers.
- $425 = 5 * 85$, where 5 is the number of anchor boxes per grid.
- $85 = 5 + 80$, where 5 is (pc, bx, by, bh, bw) and 80 is the number of classes we want to detect.

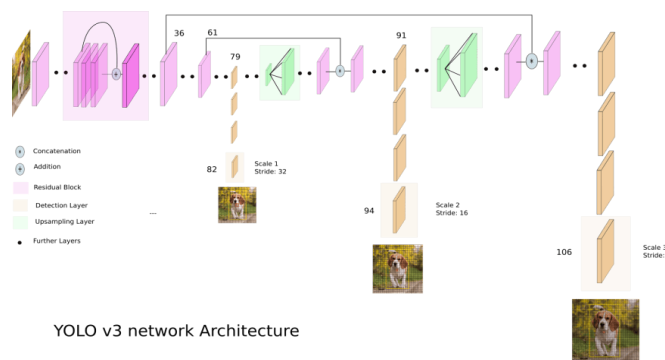


Fig. 4.1: Yolo

The output is a list of bounding boxes along with the recognized classes. Each bounding box is represented by 6 numbers (pc, bx, by, bh, bw, c). If we expand c into an 80-dimensional vector, each bounding box is represented by 85 numbers. Finally, we do the IoU (Intersection over Union) and Non-Max Suppression to avoid selecting

overlapping boxes.

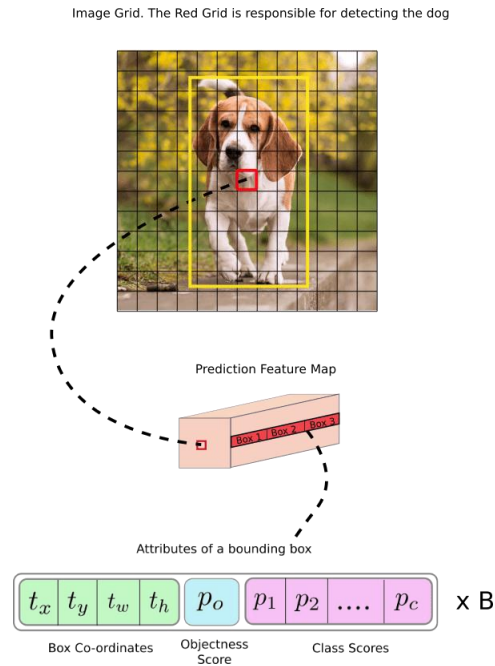


Fig. 4.2: Prediction map

YOLO v3 uses a variant of Darknet, which originally has 53 layer network trained on Imagenet. For the task of detection, 53 more layers are stacked onto it, giving us a 106 layer fully convolutional underlying architecture for YOLO v3. In YOLO v3, the detection is done by applying 1×1 detection kernels on feature maps of three different sizes at three different places in the network. The shape of detection kernel is $1 \times 1 \times (B \times (5 + C))$. Here B is the number of bounding boxes a cell on the feature map can predict, '5' is for the 4 bounding box attributes and one object confidence and C is the no. of classes. YOLO v3 uses binary cross-entropy for calculating the classification loss for each label while object confidence and class predictions are predicted through logistic regression.

4.2.1 Hyper-parameters Used

- class_threshold
- Non-Max suppression Threshold
- nput_height
- input_shape

4.2.2 CNN Architecture Of Darknet-53

Darknet-53 is used as a feature extractor. Darknet-53 mainly composed of 3 x 3 and 1 x 1 filters with skip connections like the residual network in ResNet.

4.2.3 Layers Details

YOLO makes use of only convolutional layers, making it a fully convolutional network (FCN). In YOLOv3 a deeper architecture of feature extractor called Darknet-53 is used.

4.2.4 Convolution layers in YOLOv5

It contains 53 convolutional layers which have been, each followed by batch normalization layer and Leaky ReLU activation. Convolution layer is used to convolve multiple filters on the images and produces multiple feature maps. No form of pooling is used and a convolutional layer with stride 2 is used to down sample the feature maps. It helps in preventing loss of low-level features often attributed to pooling.

CHAPTER 5

DESCRIPTION OF PROPOSED WORK

5.1 DATASET

Develop a python code for collecting different images in different situations with different backgrounds.

Use Makesense.ai to label a dataset of all photographs with a set of "known true" labels that you can use to train a yolov5 model.

When labelling your dataset for an object detection model, keep in mind the following best practices:

- Label an equal number of photographs with the features you want to identify as those without.
- Create the bounding boxes to encompass the entire person, dog, or relevant components visible in the photos.
- Label at least 5000 images of houses to train the model.
- Label images of the same resolution quality and from the same angles as those that you plan to process with the trained model.
- Limit the number of objects that you want to detect to improve model accuracy for detecting those objects.



Fig.5.1:Sample for labelling images

5.2 PROCESS MODEL

The You Only Look Once (YOLO) is computer vision models-based family that includes the model known as YOLOv5. When it comes to object detection YOLOv5 plays the best role for recognizing different sizes, orientation, etc. s, m, l and x are the four major discrepancy of YOLOv5 which are named as Small, medium, large and extra-large respectively and each of them offers accuracy rates higher respectively. Each of the variations of YOLOv5 requires different amount of time to train.

Bring into existence a fresh technique after considering the flaws of earlier attempts. The approach proposed is primarily concerned with object detection in video sequences. Without using any applications from outside sources, the system can recognize things. Whether it is live input or pre-recorded, the video format is what counts the most. The system gathers information over a specified time period for the purpose of objection detection in a video sequence. The analysis is based on the

discovered items, which are revealed in the output.

Implementation -

5.2.1 Object detection

The first step in the procedure is the collecting of images that have been carefully chosen for their superior quality, perception, and group or individual items (i.e. military weapons, missiles and drones).

Many websites (including roboflow, makesense.ai, and others) on the internet allow us to create a bounding box; roboflow, however, has some nice features like a bounding box tool, a smart polygon tool, and a polygon tool that aids in labeling the collected images for improved accuracy and validation. This clever polygon tool labels and encapsulates the things in the image. A data set is created once all the gathered images have been labeled.

Object detection features several state-of-the-art architectures that may be used on real-world datasets to recognize things with respectable accuracy. The sole requirement is that the test dataset has the same classes as the previously trained detector.

From the dataset for object detection in video sequence, some experiments are carried out. There are 61537 training images and 13387 validation images with 103 class annotations. Depending upon the guidelines training set will undergo for training and validation set will undergo for evaluation of the performance.



An altered version of the YOLOv5 (Y_v) is created specifically for this work in order to

evaluate and forecast image frames. When a video has L preceding frames, i.e., $t = 1, 2, \dots, L$, Y_v creates feature map F_t for each sequence. The number neighboring frames that are associated with the current frames which have been aggregated is denoted by τ . While it is identical to other values of τ , we explain our shell in the case of $\tau = 1$ in the bellow sections for better understanding and convenience. The Y_v module generates the associated feature flow map, $M_{q \rightarrow p}$. Two pictures in tandem I_p and I_q , where I_p represents the picture at the current frame and I_q represents the picture at the frame next to it. Using the $M_{q \rightarrow p}$, The I_p feature map will get encased by I_q feature map. The encased feature map $F_{q \rightarrow p}$ is acquired as:

$$F_{q \rightarrow p} = \tau(F_q, M_{q \rightarrow p}) \quad (1)$$

the bilinear encased function τ being used.

5.2.2 Focus layer

The focus layer is a new layer in YOLOv5 and essential module in Yolo-v5 that helps to boost the spatial resolution of feature maps at a certain point in the network. It is a convolutional layer combined with a channel shuffle layer that zooms in on minute features in the picture. The focus layer improves object detection accuracy by enhancing spatial resolution in the centre of the network. In the YOLOv5s architecture, the focus layer comes after the CSPDarknet53 backbone network and before the other detection-specific layers.

The Focus layer initially makes four duplicates of the input picture size (e.g., $3 \times 256 \times 256$). The four copies were then sampled with a step size of two (i.e., $3 \times 128 \times 128$) to create four slices. The four slices are then in-depth concatenated with an output of $12 \times 128 \times 128$, and then passed to the next convolutional layer with 32 kernel filters to generate an output of $32 \times 128 \times 128$ and the result is fed into the next convolutional layer via batch normalization and RELU as an activation function.

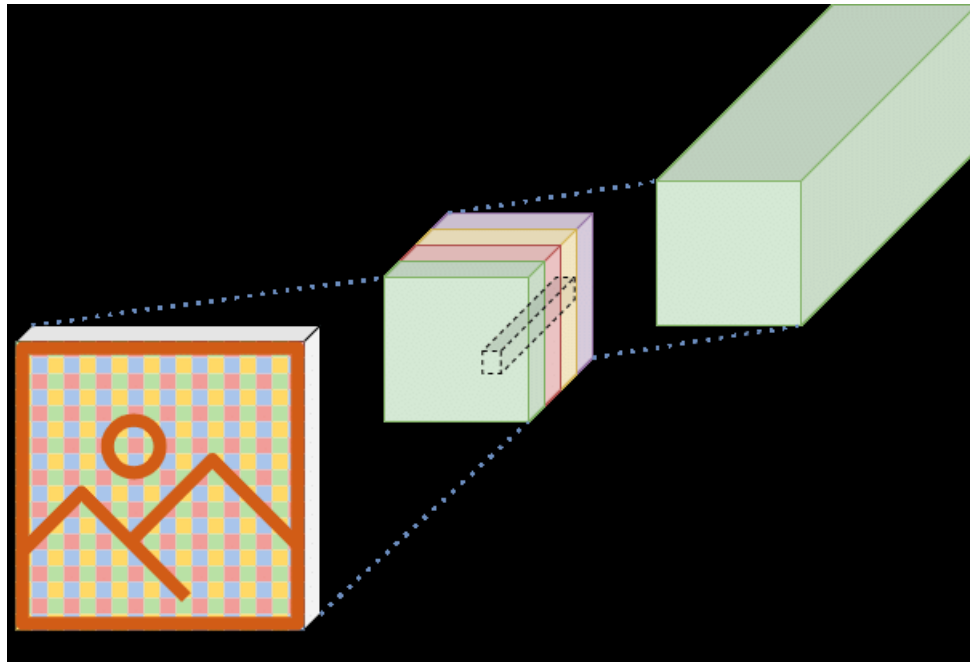


Fig.5.2:Standard focus layer of YOLO-v5 model.

5.2.3 Convolution Layers

In YOLOv5, convolutional layers are used to extract features from input images. The architecture of YOLOv5 is based on a backbone of convolutional layers, which is responsible for transforming the input image into a feature map that is then used for object detection.

The convolutional layers in YOLOv5 are typically structured as a series of blocks, each of which includes multiple convolutional layers followed by activation functions and normalization layers. The output of each block is then typically downsampled using pooling layers, which reduces the spatial dimensions of the feature maps and increases the receptive field of the network.

Convolutional layers in YOLOv5 are a key component of the network architecture used for object detection, and are responsible for extracting useful features from input images that can be used to accurately identify and localize objects of interest.

The convolutional layers in YOLOv5 play a critical role in the success of the network, contributing to its high accuracy, speed, and efficiency in object detection.

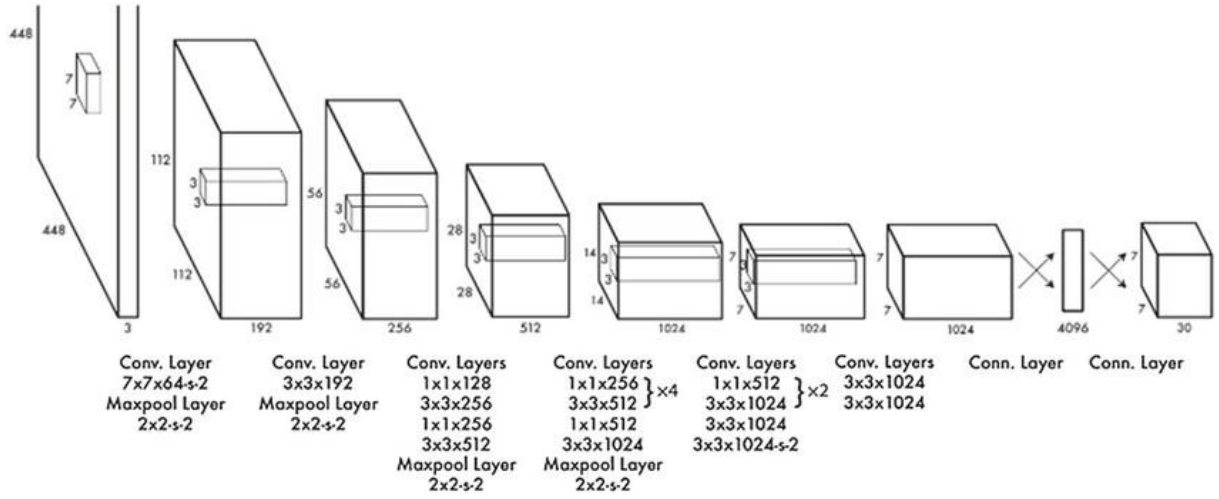


Fig.5.3: Convolution layer of YOLO-v5 model

The BottleneckCSP module in learning more about the characteristics while preserving detection speed. YOLOv5s are used to identify five sorts of surface defects: broken, hot_spot, black_border, scratch, and no_electricity. YOLOv3 shares five detection frameworks: the bipartite target detection techniques Faster-RCNN and Mask-RCNN, the classic machine learning approach SVM, and Single Shot MultiBoxDetector. This contributions can be summarised as follows:

To the best of our knowledge, we are using the YOLOv5 structure to detect flaws. YOLOv5's fast inference speed and high detection accuracy are used to achieve a combination of detection speed and accuracy on the dataset, allowing for accurate and rapid detection of various types of defects in significantly reducing missed detection of minor defects.

The structure of YOLOv5 has been enhanced and reinvented in response to the defect detection requirement. To begin, the BottleneckCSP module is used to get semantic depth information from pictures, which improves detection accuracy. Second, the addition of a detection head for small targets mitigates the detrimental effects of dramatic size shifts and improves the small target misdetection phenomena.

5.2.4 BottleneckCSP Module

$$Y = F(x_0) = x_k$$

$$= H_k(x_{k-1}, H_{k-1}(x_{k-2}), H_{k-2}(x_{k-3}), \dots, H_1(x_0), x_0) \quad (1)$$

H_k is the k^{th} layer's operator function, which typically consists of a convolution layer and an activation function. To optimise each "H" function, a "y" function was developed.

$$Y = M(x_0, T(F(x_0))), \quad (2)$$

When x_0 may be separated into two pieces along the channel, the "T" function truncates the gradient flow, and the "M" function combines the two portions. We created an information-rich feature map in order to preserve and collect more details from various sensory domains.

5.2.4.1 The effect of BottleneckCSP

The model's mAP values increased dramatically once the residual module was replaced with the BottleneckCSP module. The performance on all flaws was significantly improved, as indicated in the table below. To a significant extent, the difficulties of tiny target missed detection and detection accuracy were overcome. As a result, the BottleneckCSP module was much improved.

Table 5.1: surface layers value for broken, hot_spot, black_border, scratch, and no_electricity

Method	Broken (%)	Hot_spot (%)	Black_border (%)	Scratch (%)	No_Electricity (%)
YOLOv5s	78.5±0.05	87.8±0.04	85.4±0.02	69.3±0.06	88.0±0.04

5.2.5 SPP (spatial pyramid pooling layer)

SPP detects objects of varying sizes using a slightly different method. It adds a spatial pyramid pooling layer after the last pooling layer (after the last convolutional layer). The feature maps are spatially separated into mm bins, where "m" might be 1, 2, or 4. Then, for each channel, a maximum pool is applied to each bin. This results in a fixed-length representation that may be analysed further using FC-layers.

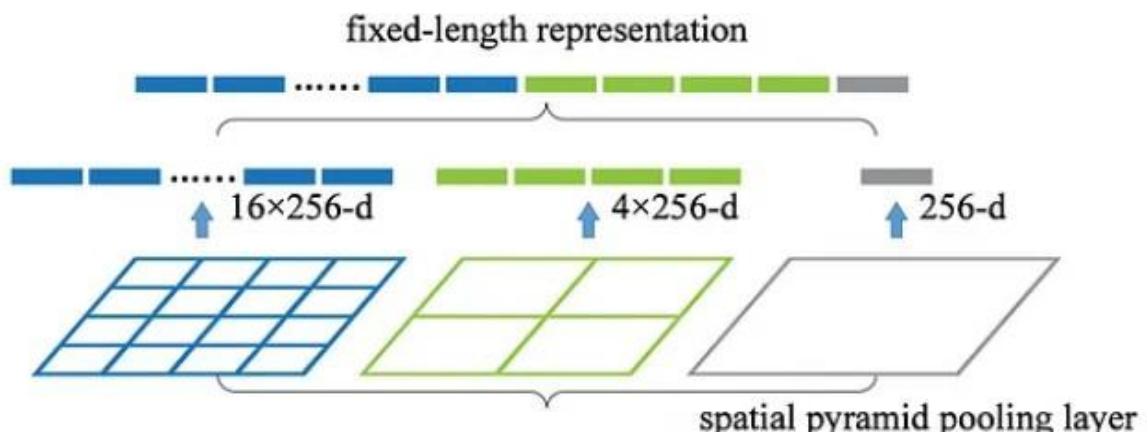


Fig.5.4: Spatial pyramid pooling layer

SPP, on the other hand, allows photos of varying sizes. Nonetheless, there exist technologies such as fully convolution networks (FCN) that do not use FC-layers and can accept pictures of varying size. This architecture is very effective for picture segmentation when spatial information is vital. As a result, converting 2-D feature maps into a fixed-size 1-D vector is not always desired for YOLO.

5.2.5.1 YOLO with SPP

The SPP in YOLO is changed to keep the output spatial dimension. A maximum pool is applied to a sliding kernel of size 1x1, 5x5, 9x9, 13x13, for example. The spatial dimension is kept. As output, the features maps from different kernel sizes are concatenated together.

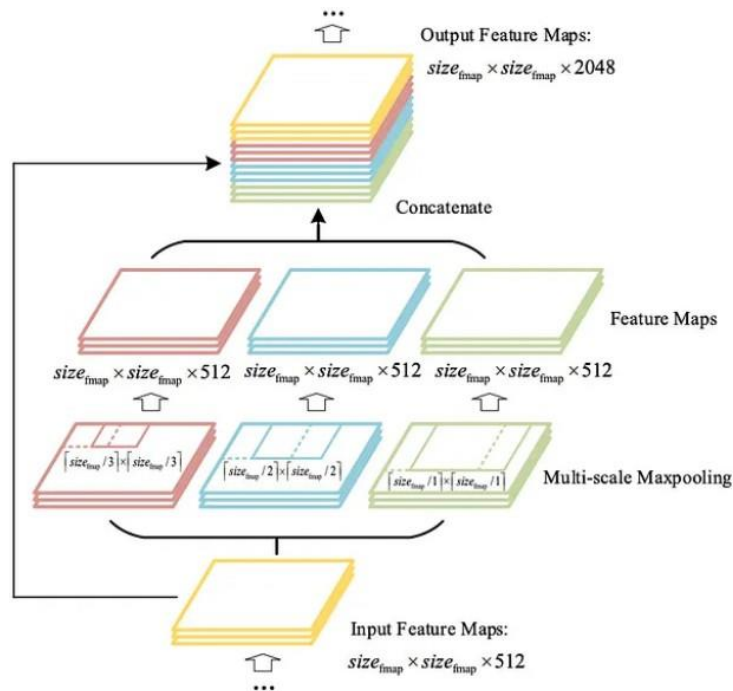


Fig.5.5: YOLO with SPP

5.3 ARCHITECTURE

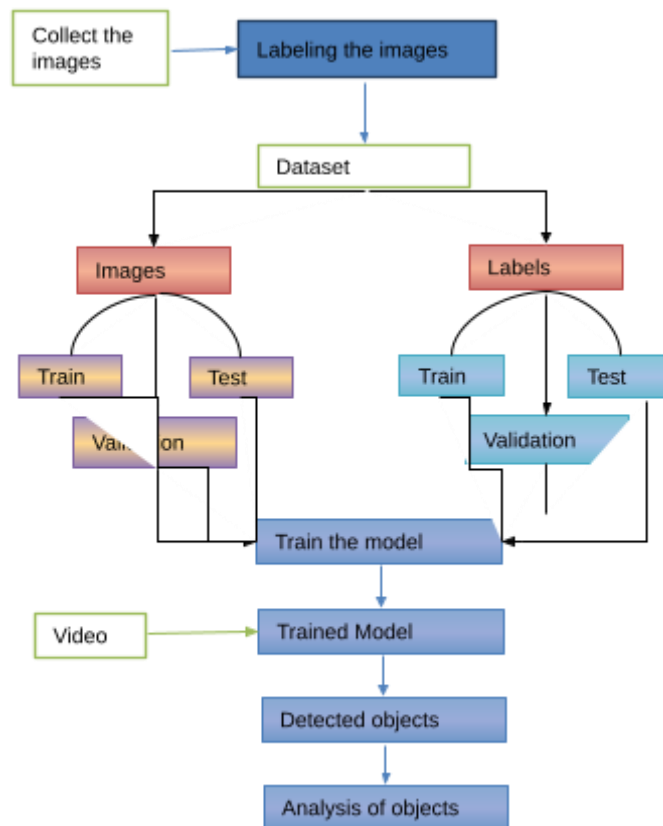


Fig.5.6: Flow diagram

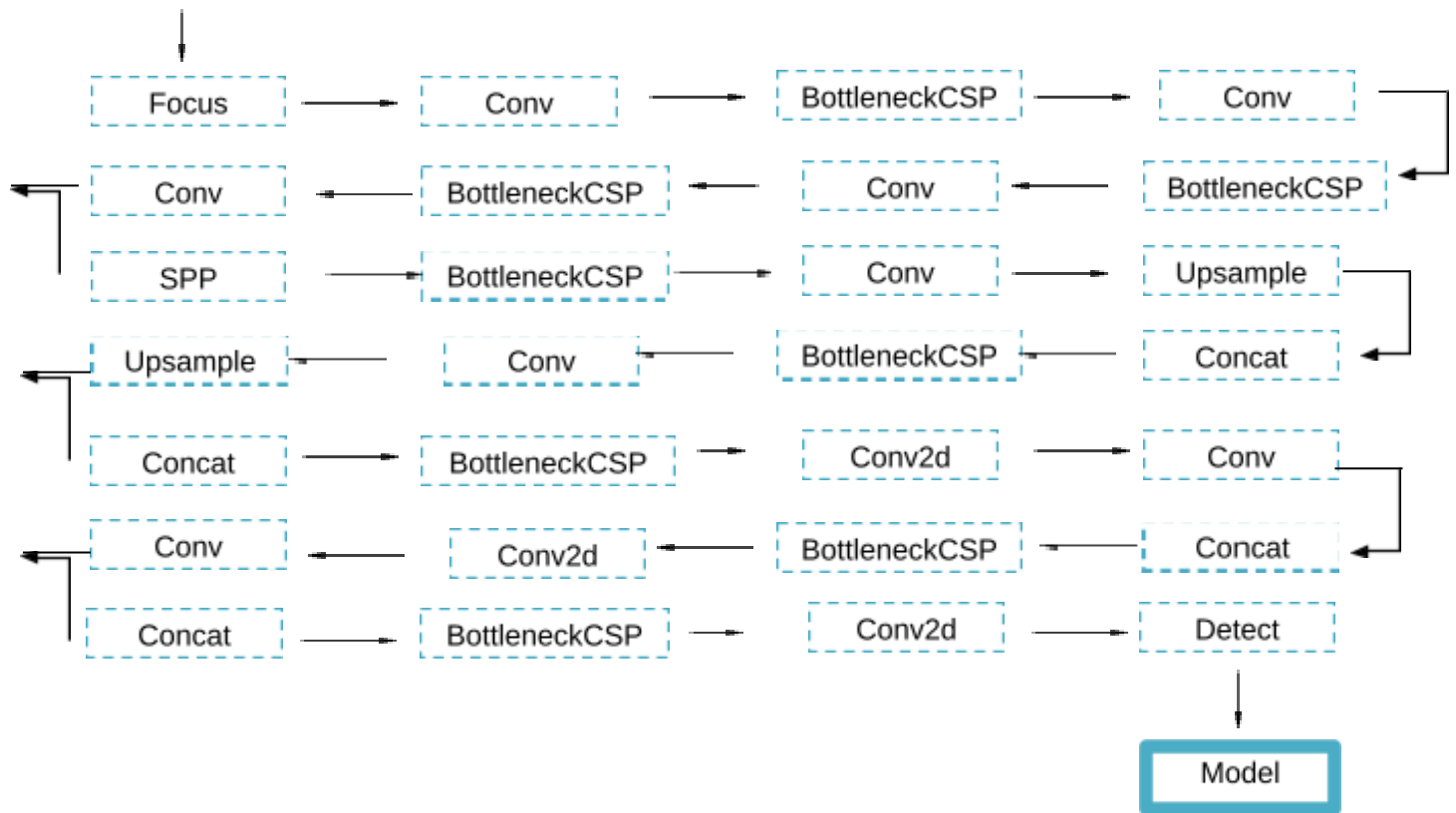


Fig.5.7: Modified YOLOv5 Architecture

$Y_v Y_v$ employs fewer anchor boxes (split the input picture into an $N \times N$ grid). This was created with the help of darknet neural networks. Now, each frame will be subjected to gradient detection using $N \times N$ grids or cells. ResNet-50 consists of 50 layers. ResNet-50 is capable of loading over a million photos from a data collection. Floating point operations it has are 3.8×10^9 which can perform better operation. "x" is defined as identity.

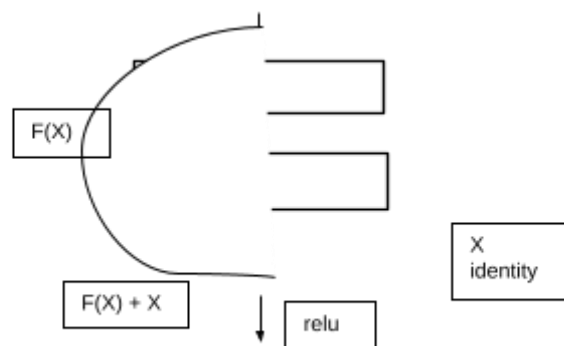


Fig.5.8: The identity will go through different weight layers with relu

Table 5.2: Summarization about the Epoch with respect to loss functions and mAP50.

30	0.0316	0.01759	0.01465	0.671
31	0.03247	0.01761	0.01135	0.762
32	0.03036	0.01686	0.01214	0.761
33	0.02997	0.01704	0.01094	0.723
34	0.02942	0.01576	0.01089	0.72
35	0.02992	0.01695	0.01288	0.698
36	0.03057	0.01607	0.01201	0.703
37	0.02749	0.01649	0.01326	0.73
38	0.02856	0.01582	0.01301	0.695
39	0.02898	0.01597	0.01223	0.733
40	0.02908	0.01628	0.01345	0.743
41	0.02804	0.01606	0.01359	0.745
42	0.02769	0.01667	0.01036	0.821
43	0.03043	0.01696	0.01111	0.681
44	0.02794	0.01598	0.01202	0.755
45	0.02694	0.01604	0.00886	0.741
46	0.02815	0.01573	0.01026	0.74
47	0.02902	0.01552	0.01234	0.675
48	0.02806	0.01549	0.01251	0.659
49	0.02756	0.01544	0.01131	0.769
50	0.02762	0.01613	0.01018	0.744
51	0.02644	0.01496	0.00973	0.73
52	0.02603	0.01445	0.01046	0.746
53	0.02754	0.01564	0.01195	0.763
54	0.02584	0.01496	0.01032	0.74
55	0.02507	0.01569	0.01122	0.768
56	0.02505	0.01524	0.012	0.775
57	0.02534	0.01469	0.00996	0.806
58	0.02633	0.01497	0.01152	0.826
59	0.02558	0.01486	0.01053	0.846
60	0.02565	0.0144	0.00922	0.878
Epoch	box_loss	obj_loss	cls_loss	mAP50
0	0.1027	0.02558	0.04088	0.0472
1	0.07733	0.02893	0.03731	0.277
2	0.06347	0.0248	0.032	0.24
3	0.06	0.02445	0.02988	0.332
4	0.0595	0.02296	0.02726	0.298
5	0.05369	0.0209	0.02454	0.433
6	0.04928	0.0221	0.02267	0.248
7	0.04784	0.01951	0.02145	0.453
8	0.04583	0.0206	0.01986	0.551
9	0.04376	0.01904	0.01813	0.623
10	0.04186	0.01851	0.01838	0.635
11	0.03858	0.01864	0.0187	0.633
12	0.03864	0.01841	0.01841	0.583
13	0.03661	0.01819	0.01728	0.647
14	0.03695	0.01756	0.01731	0.654
15	0.03727	0.01866	0.01639	0.675

16	0.0344	0.01831	0.01577	0.667
17	0.03561	0.0178	0.01525	0.679
18	0.03557	0.01775	0.01576	0.738
19	0.03355	0.01748	0.01506	0.61
20	0.03379	0.01861	0.01506	0.657
21	0.03292	0.01794	0.016	0.691
22	0.03348	0.01802	0.01404	0.626
23	0.03396	0.01659	0.01496	0.719
24	0.0321	0.01769	0.01333	0.681
25	0.03216	0.01688	0.01338	0.62
26	0.03291	0.01657	0.01578	0.608
27	0.03287	0.01664	0.0136	0.65
28	0.03135	0.01748	0.01493	0.659
29	0.0323	0.01699	0.01387	0.597

The next step is to create an interface where we may arrange the structure of input and output. The website is made up of many buttons and sorts that allow the user to quickly browse. We may either feed the model live video input or pre-recorded video input here. The model will process and analyze the video and present the detected items as a result.

The YOLO is a member of the single-shot detector family.

Network backbone: Input images will undergo extraction of essential characters which is done by Backbone. Valuable properties are extracted from an input image with the help of backbone called CSP (Cross Stage Partial Networks) in YOLOv5

5.4 ADVANTAGES

- More Reliable
- Data confidentiality
- For the calculation of each point, a sliding window effect is created.

5.5 ANALYSIS

The same video is analyzed in the model throughout the video process, where it will study the items every frame. It examines the type of thing, displays the number of objects, and determines if it is dangerous or not. This study can be presented in a variety of formats, including text, graph, table, and histogram.

5.6 PROJECT MANAGEMENT PLAN

- Step 1: Collect the dataset
- Step 2: developing a general YOLOv5 algorithm
 - Step 2.1: creating dataset
 - Step 2.2: Labeling
 - Step 2.3: training the model(YOLOv5)
- Step 3: checking the accuracy
 - Step 3.1: passing video/live stream
 - Step 3.2: checking the results
 - Step 3.3: if needed go to step 2

5.7 FINANCIAL REPORT ON ESTIMATED COSTING

The cost estimation for this project is Zero. We are using pre trained models.

CHAPTER 6

RESULT AND DISCUSSION

The enhanced the performance of YOLO models by refining the feature map and including ResNet50 in the algorithm. The best model for object identification in the video was the YOLOv5 model, which had the best mAP of 87.8% and Frame Per Second of 108.4.

The put forth an enhanced target recognition approach for video sequences based on YOLOv5. In the future, additional types of targets can be added, including armor, helicopters, sheep, pizza, and oranges. By enhancing the dataset, we further confirm the suggested model's detection ability. A self-built dataset will be used to validate the enhanced method's detection effectiveness, and it is applied to YOLOv4 and YOLOv3.

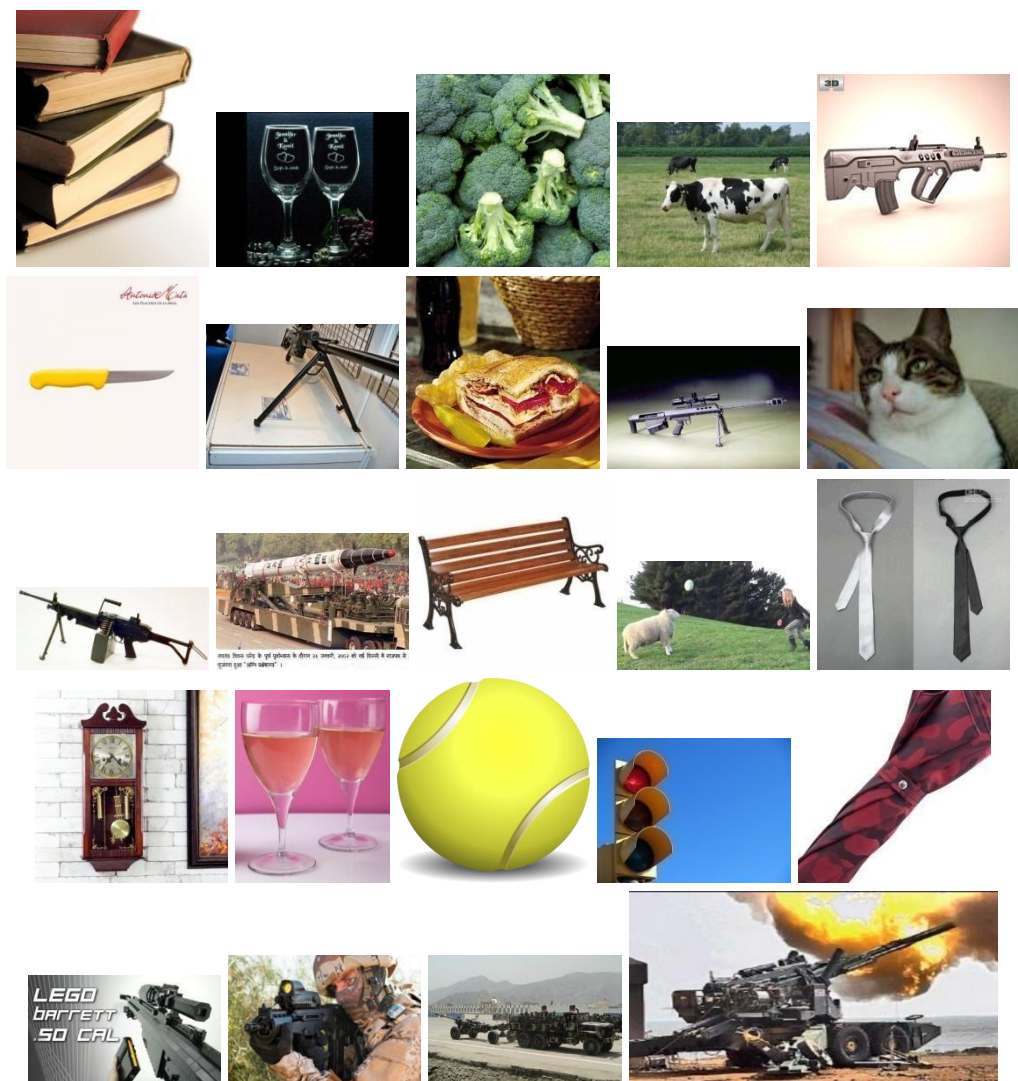


Fig.6.1:Data set



Fig.6.2:Detected objects



Fig.6.3: Detected objects



Fig.6.4: Detected objects

CHAPTER 7

CONCLUSION

Object detection is an important task in computer vision that involves identifying and localizing objects of interest within digital images or video footage. One popular approach to object detection is the YOLO (You Only Look Once) method, which seeks to detect objects in a single pass through the image or video. In this article, we will discuss objection detection in YOLOv5 and provide a brief conclusion on this topic.

YOLOv5 is the most recent version of the YOLO family of object detection models,

and it is known for its high accuracy, speed, and flexibility. YOLOv5 is capable of detecting objects of various sizes, shapes, and orientations in real-time, making it an excellent choice for applications that require fast and accurate object detection, such as autonomous driving, robotics, and surveillance.

One of the key features of YOLOv5 is its advanced object detection capabilities. YOLOv5 uses a variety of deep learning techniques, including convolutional neural networks (CNNs), to identify objects in images and videos. Specifically, YOLOv5 uses a CNN to generate a feature map of the input image, which is then used to predict the presence and location of objects in the image.

Another important aspect of object detection in YOLOv5 is the use of anchor boxes. Anchor boxes are a set of predefined boxes that are used to represent different object sizes and shapes. YOLOv5 uses anchor boxes to improve the accuracy of object detection by allowing the model to predict the location and size of objects more accurately.

In addition to the above features, YOLOv5 also incorporates a number of other advanced techniques, such as focal loss, multi-scale training, and data augmentation, all of which contribute to its superior object detection capabilities.

In conclusion, YOLOv5 is a powerful and versatile object detection model that can be used to detect objects of various sizes, shapes, and orientations in real-time. Its advanced object detection capabilities, including the use of CNNs, anchor boxes, and other advanced techniques, make it an excellent choice for a range of applications, from autonomous driving to surveillance and beyond. If you are looking for a fast and accurate object detection model, YOLOv5 is definitely worth considering.

7.1 FUTURE WORK

- Object detection is one of the fundamental tasks in the field of computer vision that involves identifying objects within an image or a video stream. YOLOv5 is a deep learning-based object detection model that has gained popularity due to its speed and accuracy. However, there is still room for improvement in the area of object detection within YOLOv5.
- One area of future work for object detection in YOLOv5 is the detection of

small or occluded objects. One of the challenges in object detection is identifying small or partially visible objects within an image. One possible solution is to use a multi-scale approach, where the size of the input images is varied during the training process. This approach can help to improve the detection of small objects within an image. Similarly, an attention mechanism can also be incorporated that selectively focuses on small or occluded objects.

- Another area of future work is improving the detection of objects in challenging environments. Environmental factors such as low light, shadows, and reflections can affect the detection of objects. In this context, improved image preprocessing techniques and the incorporation of additional sensors such as thermal or radar can help to improve detection in challenging environments.
- Additionally, incorporating information from multiple modalities can help to improve the accuracy of objection detection in YOLOv5. For example, using textual descriptions or context can help to disambiguate the identities of objects within the image. Furthermore, incorporating temporal information can help to track objects across multiple frames, improving the detection of objects with complex motion patterns.
- Finally, improving the efficiency of the computation within YOLOv5 can also be an area of future work. While YOLOv5 is already known for its speed, further improvements in computation can help to make the algorithm more efficient, even on low-power devices.
- In conclusion, there are still areas within YOLOv5 object detection that can be improved upon. Future work in these areas can not only increase the accuracy and applicability of the model but can also help to address current challenges in the field of computer vision.

7.2 RESEARCH ISSUES

- Object detection is a challenging task in the field of computer vision, and it has numerous applications in different domains, such as robotics, surveillance systems, autonomous driving, and face recognition. YOLOv5 is a

state-of-the-art object detection network that utilizes deep learning algorithms for detecting and classifying objects in real-time. However, like any other object detection network, YOLOv5 has its own research issues, which need to be addressed to improve its performance and accuracy. In this article, we will discuss some of the research issues in YOLOv5 object detection.

- The first research issue in YOLOv5 object detection is the detection of small objects. YOLOv5 has a relatively low resolution of feature maps, which makes it difficult to detect small objects accurately. To address this issue, researchers have proposed several techniques, such as using multiscale feature fusion, feature pyramid networks, and spatial pyramid pooling.
- The second research issue is the detection of occluded objects. YOLOv5 has a limited understanding of occlusions, which can lead to false positives or negatives. For example, if a person is obstructed by an object, YOLOv5 may not detect the person correctly. To address this issue, researchers have proposed several techniques, such as using context-based reasoning, attention mechanisms, and occlusion-awareness.
- The third research issue is the detection of objects with complex shapes. YOLOv5 has a simplified architecture that uses bounding boxes to represent objects. However, some objects have complex shapes that cannot be accurately represented by bounding boxes. To address this issue, researchers have proposed several techniques, such as using deformable convolutional networks, shape-based representations, and contour detection.
- The fourth research issue is the detection of rare or novel objects. YOLOv5 is trained on large datasets that contain common objects. However, rare or novel objects may be challenging to detect because they are not present in the training data. To address this issue, researchers have proposed several techniques, such as using transfer learning, meta-learning, and few-shot learning.
- In conclusion, YOLOv5 object detection is a complex task that requires addressing several research issues to improve its performance and accuracy. The above-discussed issues are just a few examples, and there are many more research issues that need to be addressed to advance the field of object detection.

7.3 IMPLEMENTATION ISSUES

- Object detection using YOLOv5 is a popular technique that allows for the detection and classification of objects in real-time video streams. However, there are several implementation issues that need to be considered when using this technique.
- One of the primary issues with implementing object detection using YOLOv5 is the need for sufficient computing resources. YOLOv5 requires a significant amount of processing power to run complex deep learning algorithms, and this can lead to performance issues on lower-end hardware. However, advancements in hardware technology have made it possible to run YOLOv5 on devices with lower specifications, such as smartphones and embedded systems.
- Another implementation issue with YOLOv5 is the need for a large dataset of training images. Because it is a deep learning algorithm, YOLOv5 relies on a diverse range of images to learn how to identify objects in different environments and scenarios. This requires a significant amount of time and resources to gather and label a sufficient number of images for training.
- Additionally, it is important to consider the trade-off between accuracy and speed when implementing object detection using YOLOv5. The faster the algorithm, the less accurate it may be, and vice versa. This trade-off is particularly relevant when implementing object detection in real-time video streams, where accuracy and speed are both important factors to consider. Finally, it is important to consider the ethical implications of object detection using YOLOv5. The algorithm relies on a dataset of images that may not be representative of all individuals or groups, which can lead to bias and discrimination. It is important to consider these issues and take steps to mitigate them, such as using data that is more diverse and representative of different groups.
- In conclusion, while YOLOv5 is a powerful technique for object detection, there are several implementation issues to consider, such as computing resources, dataset size, accuracy vs. speed trade-offs, and ethical implications. By taking these issues into account, it is possible to implement

object detection using YOLOv5 effectively while ensuring fairness and accuracy.

REFERENCES

- [1] Isa, I. S., Rosli, M. S. A., Yusof, U. K., Maruzuki, M. I. F., & Sulaiman, S. N. (2022). Optimizing The Hyperparameter Tuning of YOLOv5 For Underwater Detection. *IEEE Access*.
- [2] Luo, Shan, and Jihong Liu. "Research on Car License Plate Recognition Based on Improved YOLOv5m and LPRNet." *IEEE Access* 10 (2022): 93692-93700.
- [3] Wang, Chenglong, Ziran Zhou, and Zhiming Chen. "An Enhanced YOLOv4 Model with Self-Dependent Attentive Fusion and Component Randomized Mosaic

Augmentation for Metal Surface Defect Detection." *IEEE Access* 10 (2022): 97758-97766.

[4] Kong, Lingren, Jianzhong Wang, and Peng Zhao. "YOLO-G: A Lightweight Network Model for Improving the Performance of Military Targets Detection." *IEEE Access* (2022).

[5] Yang, Kuihe, and Ziyang Song. "Deep Learning-Based Object Detection Improvement for Fine-Grained Birds." *IEEE Access* 9 (2021): 67901-67915.

[6] Zhou, Li-Qun, Peng Sun, and Jin-Chun Piao. "A Novel Object Detection Method in City Aerial Image Based on Deformable Convolutional Networks." *IEEE Access* 10 (2022): 31455-31465.

[7] Li, H., Deng, L., Yang, C., Liu, J., & Gu, Z. (2021). Enhanced YOLO v3 tiny network for real-time ship detection from visual image. *IEEE Access*, 9, 16692-16706.

[8] Zhao, J., Hao, S., Dai, C., Zhang, H., Zhao, L., Ji, Z., & Ganchev, I. (2022). Improved Vision-Based Vehicle Detection and Classification by Optimized YOLOv4. *IEEE Access*, 10, 8590-8603.

[9] Li, Hui, Lizong Liu, Jun Du, Fan Jiang, Fei Guo, Qilong Hu, and Lin Fan. "An Improved YOLOv3 for Foreign Objects Detection of Transmission Lines." *IEEE Access* 10 (2022): 45620-45628.

[10] Wang, Zhuang-Zhuang, Kai Xie, Xin-Yu Zhang, Hua-Quan Chen, Chang Wen, and Jian-Biao He. "Small-Object Detection Based on YOLO and Dense Block via Image Super-Resolution." *IEEE Access* 9 (2021): 56416-56429.

[11] Hong, Zhonghua, Ting Yang, Xiaohua Tong, Yun Zhang, Shenlu Jiang, Ruyan Zhou, Yanling Han, Jing Wang, Shuhu Yang, and Sichong Liu. "Multi-scale ship detection from SAR and optical imagery via a more accurate YOLOv3." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021): 6083-6101.

[12] An, Qilin, Kai Wang, Zhongyang Li, Chengyuan Song, Xiuying Tang, and Jian Song. "Real-Time Monitoring Method of Strawberry Fruit Growth State Based on YOLO Improved Model." *IEEE Access* 10 (2022): 124363-124372.

[13] Zhang, L., Wang, M., Liu, K., Xiao, M., Wen, Z., & Man, J. (2022). An Automatic Fault Detection Method of Freight Train Images Based on BD-YOLO. *IEEE Access*, 10, 39613-39626.

- [14] Lin, Cheng-Jian, and Jyun-Yu Jhang. "Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks." *IEEE Access* 10 (2022): 14120-14133.
- [15] Li, S., Li, Y., Li, Y., Li, M., & Xu, X. (2021). YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. *IEEE Access*, 9, 141861-141875.

APPENDIX

A.

SOURCE CODE

```
from re import DEBUG, sub
from flask import Flask, render_template, request, redirect, send_file,
url_for
from werkzeug.utils import secure_filename, send_from_directory
```



```

import os

import subprocess

app = Flask(__name__)

uploads_dir = os.path.join(app.instance_path, 'uploads')

os.makedirs(uploads_dir, exist_ok=True)

@app.route("/")
def index():
    return render_template('index.html')

@app.route("/main")
def main():
    return render_template('main.html')

@app.route("/upload")
def upload():
    return render_template('upload.html')

@app.route("/d", methods=['POST'])
def d():
    if 'file' not in request.files:
        flash('No file part')
        return render_template('upload.html')
    video = request.files['file']
    if video.filename == "":
        flash('No image selected')

```

```

        return render_template('upload.html')

    if video:
        video.save(os.path.join(uploads_dir, secure_filename(video.filename)))
        print(video)
        subprocess.run("dir", shell=True)
        subprocess.run(['py', 'detect.py', '--source', os.path.join(uploads_dir,
        secure_filename(video.filename))], shell=True)
        obj = secure_filename(video.filename)
        return render_template('download.html')

```

```

@app.route("/detect", methods=['POST'])
def detect():
    if not request.method == "POST":
        return

    video = request.files['video']
    video.save(os.path.join(uploads_dir, secure_filename(video.filename)))
    print(video)
    subprocess.run("dir", shell=True)
    subprocess.run(['py', 'detect.py', '--source', os.path.join(uploads_dir,
    secure_filename(video.filename))], shell=True)

    # return os.path.join(uploads_dir, secure_filename(video.filename))
    obj = secure_filename(video.filename)
    return obj

```

```

@app.route("/opencam", methods=['GET'])
def opencam():
    print("here")
    subprocess.run(['py', 'detect.py', '--source', '0'], shell=True)

```

```
return "done"
```

```
@app.route('/return-files', methods=['GET'])
```

```
def return_file():
```

```
    obj = request.args.get('obj')
```

```
    loc = os.path.join("runs/detect", obj)
```

```
    print(loc)
```

```
    try:
```

```
        return send_file(os.path.join("runs/detect", obj),
```

```
attachment_filename=obj)
```

```
        # return send_from_directory(loc, obj)
```

```
    except Exception as e:
```

```
        return str(e)
```

```
# @app.route('/display/<filename>')
```

```
# def display_video(filename):
```

```
#     #print('display_video filename: ' + filename)
```

```
#     return redirect(url_for('static/video_1.mp4', code=200))
```

Train.py

```
import argparse
```

```
import math
```

```
import os
```

```
import random
```

```
import subprocess
```

```
import sys
```

```
import time
```

```
from copy import deepcopy
```

```
from datetime import datetime
```

```

from pathlib import Path

import numpy as np
import torch
import torch.distributed as dist
import torch.nn as nn
import yaml

from torch.optim import lr_scheduler
from tqdm import tqdm

FILE = Path(__file__).resolve()
ROOT = FILE.parents[0] # YOLOv5 root directory
if str(ROOT) not in sys.path:
    sys.path.append(str(ROOT)) # add ROOT to PATH
ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative

import val as validate # for end-of-epoch mAP
from models.experimental import attempt_load
from models.yolo import Model
from utils.autoanchor import check_anchors
from utils.autobatch import check_train_batch_size
from utils.callbacks import Callbacks
from utils.dataloaders import create_dataloader
from utils.downloads import attempt_download, is_url
from utils.general import (LOGGER, TQDM_BAR_FORMAT, check_amp,
    check_dataset, check_file, check_git_info,
    check_git_status, check_img_size, check_requirements, check_suffix,
    check_yaml, colorstr,
    get_latest_run, increment_path, init_seeds, intersect_dicts,
    labels_to_class_weights,
    labels_to_image_weights, methods, one_cycle, print_args, print_mutation,
    strip_optimizer,
    yaml_save)

```

```

from utils.loggers import Loggers
from utils.loggers.comet.comet_utils import check_comet_resume
from utils.loss import ComputeLoss
from utils.metrics import fitness
from utils.plots import plot_evolve
from utils.torch_utils import (EarlyStopping, ModelEMA, de_parallel,
select_device, smart_DDP, smart_optimizer,
smart_resume, torch_distributed_zero_first)

LOCAL_RANK = int(os.getenv('LOCAL_RANK', -1)) #
https://pytorch.org/docs/stable/elastic/run.html
RANK = int(os.getenv('RANK', -1))
WORLD_SIZE = int(os.getenv('WORLD_SIZE', 1))
GIT_INFO = check_git_info()

def train(hyp, opt, device, callbacks): # hyp is path/to/hyp.yaml or hyp
dictionary
save_dir, epochs, batch_size, weights, single_cls, evolve, data, cfg, resume,
noval, nosave, workers, freeze = \
    Path(opt.save_dir), opt.epochs, opt.batch_size, opt.weights,

```

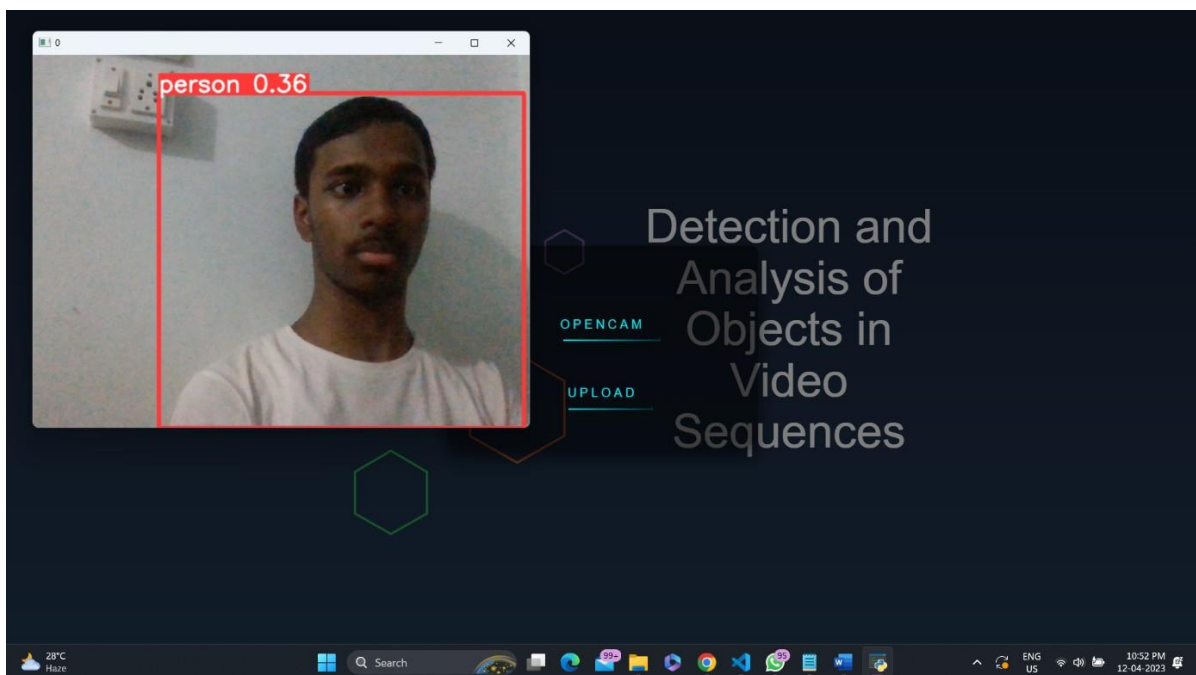
B. SCREENSHOTS

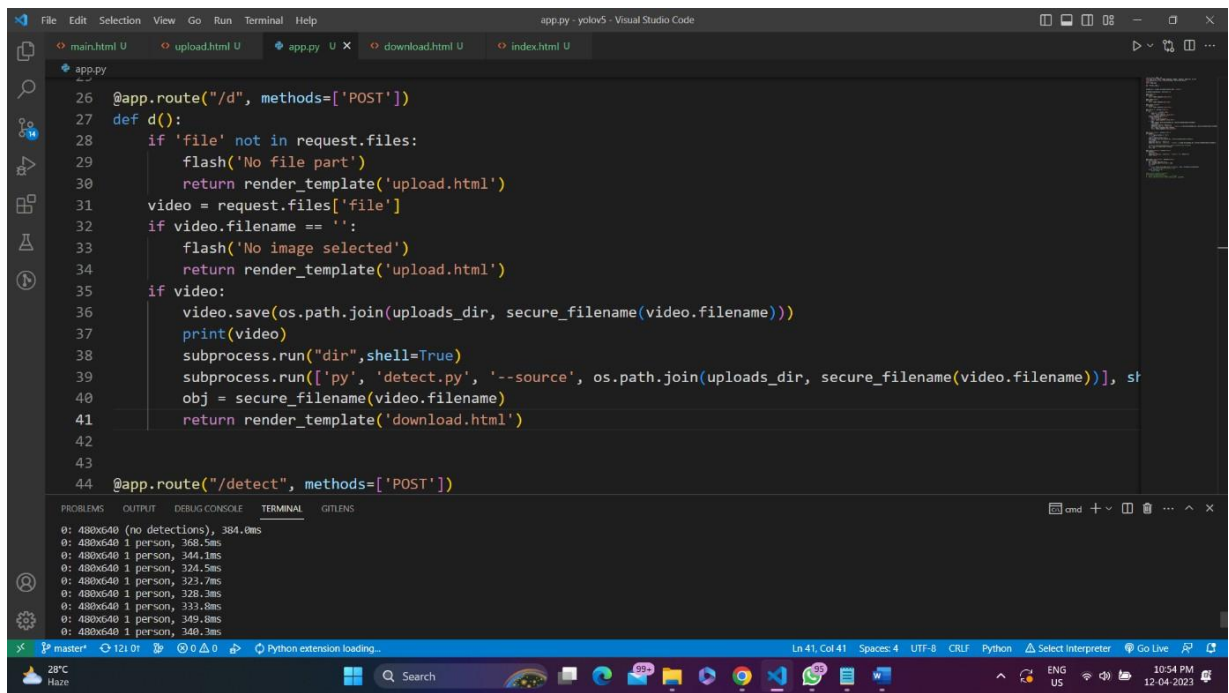


Detection and Analysis of Objects in Video Sequences



Detection and Analysis of Objects in Video Sequences





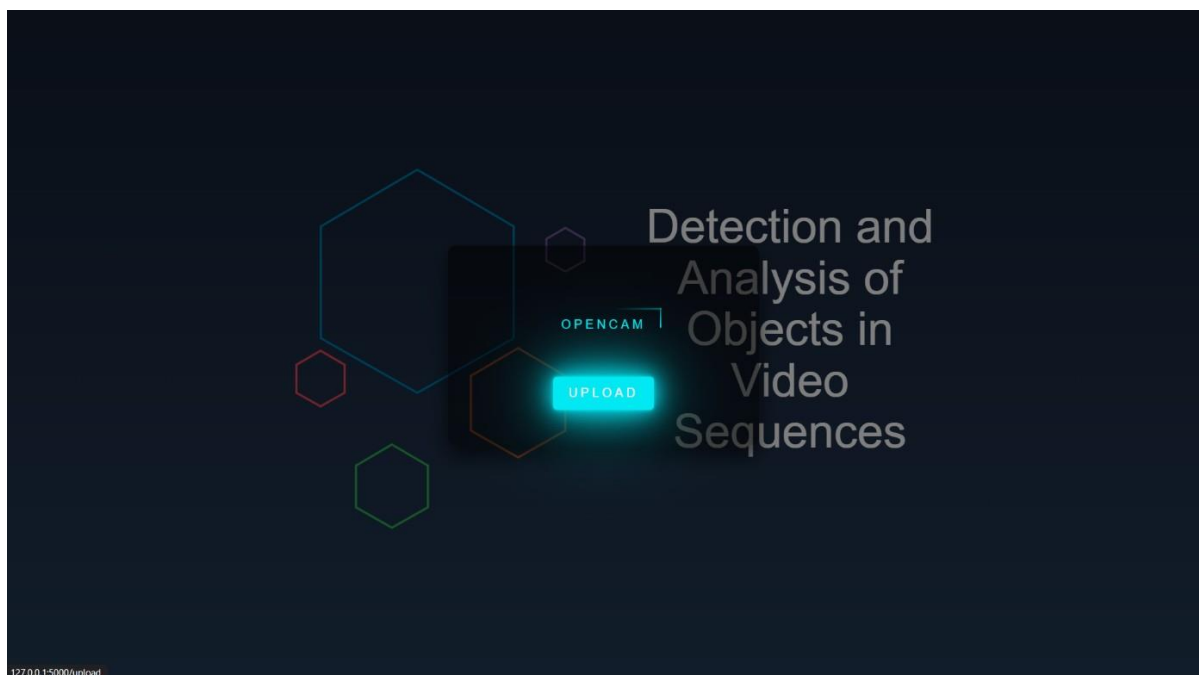
```
26 @app.route("/d", methods=['POST'])
27 def d():
28     if 'file' not in request.files:
29         flash('No file part')
30         return render_template('upload.html')
31     video = request.files['file']
32     if video.filename == '':
33         flash('No image selected')
34         return render_template('upload.html')
35     if video:
36         video.save(os.path.join(uploads_dir, secure_filename(video.filename)))
37         print(video)
38         subprocess.run("dir", shell=True)
39         subprocess.run(['py', 'detect.py', '--source', os.path.join(uploads_dir, secure_filename(video.filename))], st
40     obj = secure_filename(video.filename)
41     return render_template('download.html')
42
43
44 @app.route("/detect", methods=['POST'])
```

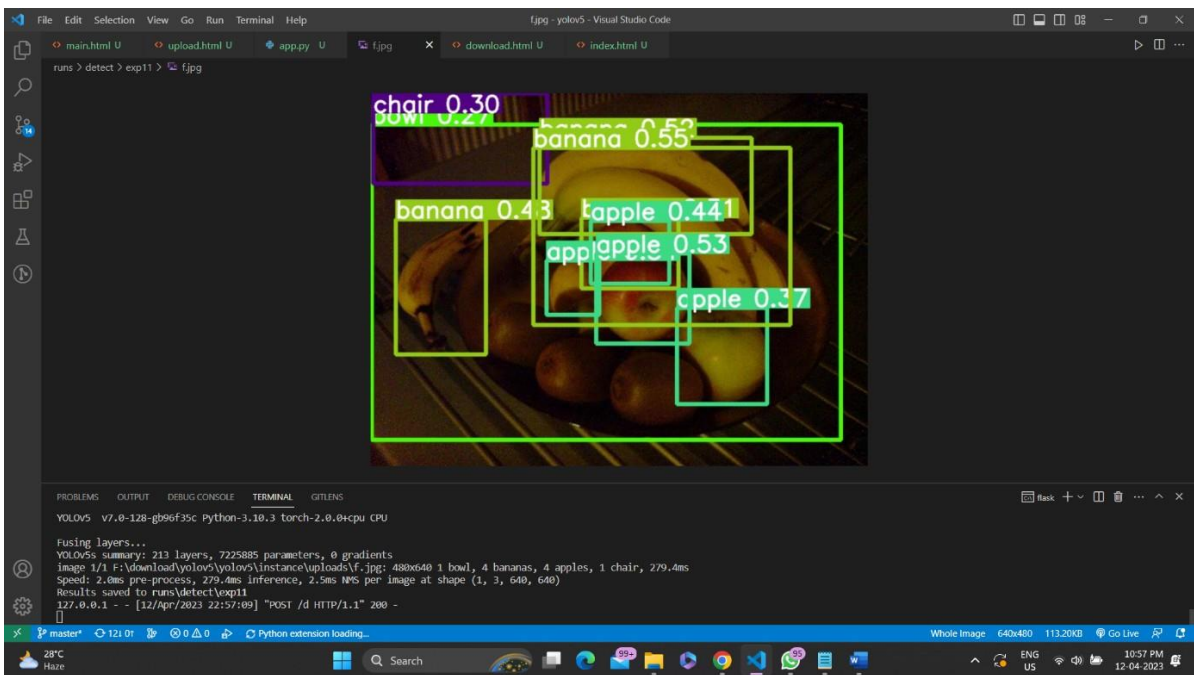
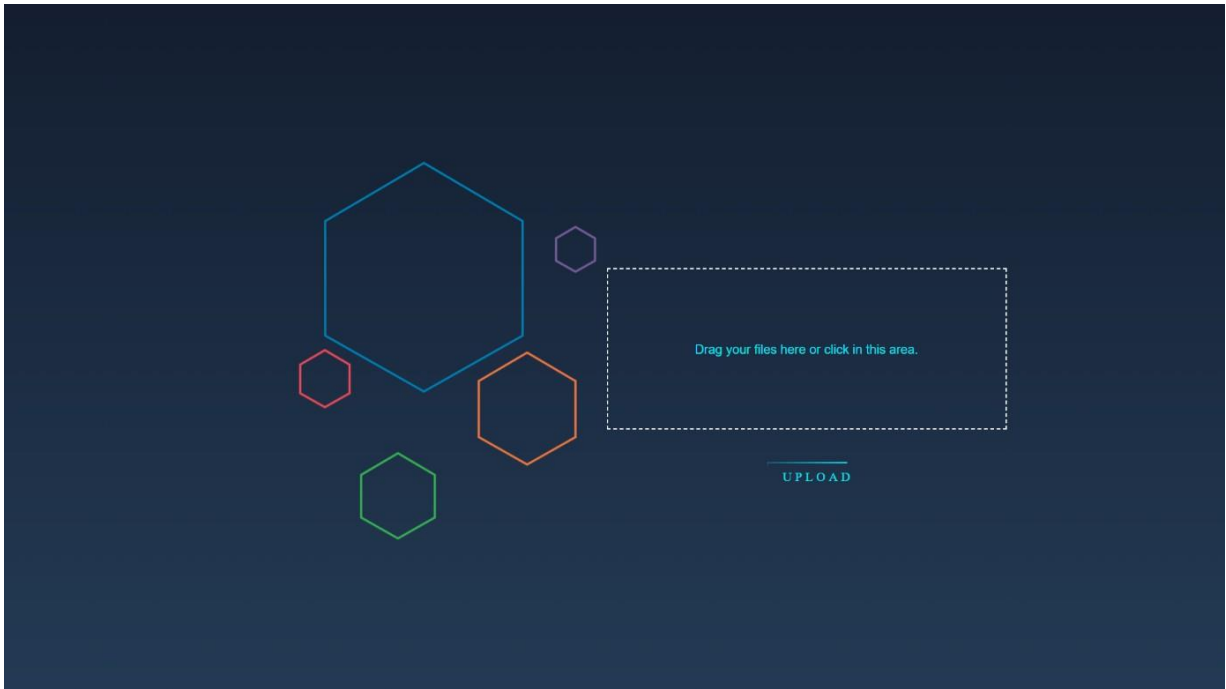
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GIT LENS

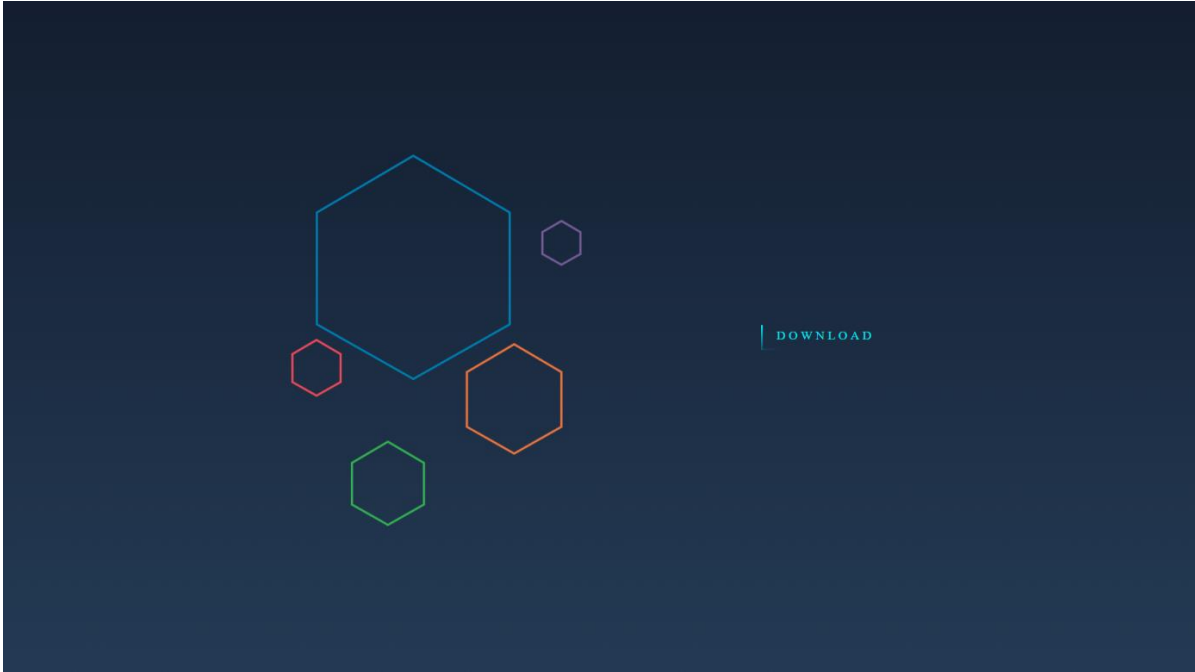
0: 480x640 (no detections), 384.0ms
0: 480x640 1 person, 368.5ms
0: 480x640 1 person, 344.1ms
0: 480x640 1 person, 324.5ms
0: 480x640 1 person, 323.7ms
0: 480x640 1 person, 326.3ms
0: 480x640 1 person, 333.8ms
0: 480x640 1 person, 349.8ms
0: 480x640 1 person, 340.3ms

Ln 41, Col 41 Spaces: 4 UTF-8 CRLF Python Select Interpreter Go Live

28°C Haze







Detection and Analysis of Objects in Video Sequences

Kolla Om Vivek
Computer Science And Engineering
Sathyabama Institute Of Science And
Technology
Chennai, India
omvivekk@gmail.com

Kolla Om Vitesh
Computer Science And Engineering
Sathyabama Institute Of Science And
Technology
Chennai, India
omvitesh@gmail.com

Dr.D. Usha Nandini
Computer Science And Engineering
Sathyabama Institute Of Science And
Technology
Chennai, India
usha.cse@sathyabama.ac.in

Dr.PrinceMary S Computer
Science And Engineering
Sathyabama Institute Of Science And
Technology
Chennai,
Indiaprincemary.cse@sathyabama.ac.in

Dr.Mercy Paul SelvanComputer
Science And Engineering
Sathyabama Institute Of Science And
Technology
Chennai,
Indiamercypaulselvan.cse@sathyabama
.ac.in

Abstract—Identifying and analyzing elements in the video can be difficult due to changes in lighting, the appearance of the subject, and comparable off-target elements in the background. In this study, we used YOLOv5 to extract and classify objects in the video. A dataset called "Detection" was created using Python approaches for detecting objects. The detection dataset contains many categories of "x" photographs. The YOLOv5 models were adjusted and refined to identify real-time items such as humans, dogs, rifles, etc. An annotated Detection dataset was used to train YOLOv5, which has been fine-tuned for faster performance and improved detection accuracy. The created model was used to determine bounding boxes for the objects in the video. Additionally, each object is identified by its name and highlighted with a different color, making object recognition easier.

Keywords—YOLO,R-CNN, CNN, VGG, DARKNET-53

I. INTRODUCTION

In many applications, including visual surveillance, PPE detection, anomaly and defect identification, traffic monitoring and road maintenance, human-computer interaction, ADA compliance, and others, object recognition and analysis in video frames are now becoming a very important area. The primary purpose of object detection is to recognise and estimate the class of an object from a given picture in a video sequence. Computer vision enables computers and software to obtain digital information about an image or video. Encoding objects play a vital role in computer vision technologies. Therefore, detection accuracy with the basal false positive rate is essential. However, real-time detection in video sequences is technically difficult and time-consuming to do on a series of images due to several factors. One of them is that the images are made

up of a large amount of labelled data that must be processed through complex operations. Since the advent of the video coding standard, analysis has become particularly important in allowing content-based image coding in this field. The advent of learning technology has changed centuries of object detection and identification methods, providing comprehensive information about detected objects.

owing to the variety of uses, object detection has recently gained prominence. For example, in the field of anomaly and defect detection, the ability to automatically identify and classify anomalies or defects can help improve quality assurance consistency as well as the efficiency and effectiveness of production assembly. All data about flaws and abnormalities is retained in the system by the detecting tools. The machine can draw inferences from it and improve its detecting capabilities over time. Whereas in the case of the classic/typical defect and anomaly detection system, the efficacy of quality perception might plummet with each staff shift - resulting in expenditures. For example, in the realm of human activities, People Counting- Counting people automatically have plural uses in smart cities and public spaces. Tracking the overall number of attendees to events or public attractions is crucial for safety and planning. In transportation systems, object detection may be used to determine the most popular bus or subway stations and offer statistics on route and capacity adjustments.

The capacity to precisely extract characteristics and patterns from the visual input is essential for object recognition to be successful. Object stability, or the capacity to recognise an element under various viewing circumstances, is a crucial component of object recognition. Object orientation, lighting, and object diversity (size, colour, and other differences in the catalog) are a few of these many situations. The visual system must be able to infer the similarity of an object description across different retinal views and descriptions to achieve object consistency. Edge

detection and texture analysis are two examples of low-level image processing methods that are frequently used in conjunction with advanced machine learning algorithms like support vector machines (SVM) and You Only Look Once version 5 (YOLOv5). It is feasible to attain high levels of accuracy and resilience in real-world circumstances by training these algorithms on huge datasets of annotated activity samples.

II. LITERATURE SURVEY

Object detection and analysis in video sequences have been the subject of numerous studies in recent years. Research on object detection is compiled in this section.

Rosli and others proposed a YOLOv5 model by optimizing the hyperparameter for underwater detection [1]. They trained the YOLOv5 model with a dataset containing various bright and blur images to check the performance of the algorithm based on quantitative results and frame rate. Based on momentum and learning rate, the feature extraction phase's hyperparameter was adjusted and further enhanced by ADAM in YOLOv5. The model's accuracy increased by 98.6%, and its frame rate increased by 106 frames per second as a result. ADAM has a learning rate and momentum of 0.0001 and 0.99, respectively, achieving greater accuracy for detecting objects underwater. Shan Luo and Jihong Liu [2] proposed an improved YOLOv5m and LPRNet model for car licence plate recognition. They used K-means++ and the DIOU loss function to improve YOLOV5M while removing the 20 x 20 feature map. After that, they went to LPRNet, which used to recognise characters on the licence plate. We got the model by combining the improved YOLOv5m and LPRNet. The accuracy improved to 99.47%. By adding three more components to the existing YOLOV4 model, Chenglong Wang presented an improved YOLOV4 model for metal surface defect identification [3]. The first component among the three is self-dependent attentive fusion (SAF), which helps improve inter-path and feature fusion, followed by component-randomized mosaic augmentation, which helps find the over-transformed image, and the last one is perturbation-agnostic, which helps with regularization. Based on the proposed model, we found that YOLOv4 had 6.51% and YOLOv5 had 3.76% after validating it. Lingren Kong proposed a novel-based model called Yolo-g, it is a compact model for enhancing military object detection performance [4]. When we're discussing militaries, it's all about target tracking and analysing the battlefield situation. It's already known that with the help of deep learning techniques, it's not possible to create a good model for military targets. So, Lingren Kong took the already existing YOLOv3 and added GhostNet, a compact CNN to the extracting features network. which helps in the military's ability to detect targets quickly and accurately. With the help of the DIOU loss function, he redesigned the loss function for target detection. Comparison to the original YOLOv3 algorithm, experimental findings demonstrate that our

technique increases the detection rate by 25.9 images per sec and the MAP by 2.9%, respectively and the suggested model's size is reduced by 1/6 of YOLOv3's.

Many Detecting objects in video sequence methods emerged from image-based detectors since video is made up of frames of pictures. Detecting and analyzing objects in video has the virtue of not being altered by environmental factors, and it is a prime spot for research in detection of objects in video sequence. The main works of detecting and analyzing objects in a video sequence are as follows: (1) An accomplishment of the spatiotemporal circumstances that form the setting for an information of broadcasting of moving visual images to strengthen accuracy; (2) tackle unusual and difficult video problems using local or global a vital piece of information (e.g., occlusion, unusual posture, etc.) and (3) To boost performance, decrease the processing of superfluous sections between video frames.

There are two types of Detectors that we can use, they are described as follows:

Image-Based: anchor-based (e.g., YOLOv5) and anchor-free are the two classified types of present state-of-the-art [SOTA] or significant detectors. The majority of them are built on a solid capability of extracting picture information. They have a diverse set of applications and settings. When used directly for video detection, these detectors tend to perform badly. In lieu, they don't take use of video's distinctive & vital content. On the flip side, they lack specific enhancement for low-quality frames. To a wide range of detectors, post-processing approach is applicable. While employing video spatiotemporal context to increase detector accuracy. It lowers the rate of missed detection, increases the rate of erroneous detection, and enhances resilience.

Video-Based: Several technological approaches to video object detection have recently been developed. Following the flow-based concept, DFF and THP are proposed by X Zhu. Convolution networks and optical combination is the approach used; precisely, in order to avoid costly computations for non-important frames, the model conducts sophisticated intricacy and feature aggregation on a few critical sets (or portions in the frame) and computation of flow of optics on parts. It should be noted that the author of THP [10] recommends a strategy based on learning for distinguishing important areas from non-critical zones, this prompted us to develop a learning-based method for determining the similarity of thoughts.

They are broadly classified into two types: box and feature are the two-level methods. They use temporal information on the box level in the box level approaches. The anticipated bounding boxes each frame is connected in a sequence along the time axis. The original score dispersion is then used to re-score these linked boxes.

Table 1

Summarization about the existing system, methodology, and results

Author	Data	Algorithm	Result
Iza Sazanits Isa	Underwater Detection	Yolov5	90%
Jihong Liu	License Plate Recognition	YOLOv5 m	87%
Chenglong Wang	Metal Surface Defect	YOLOv4	82%
Jianzhong Wang	Military Targets	YOLO-G	84%
Ziying Song	Fine-Grained Birds	YOLOv3	79%
Li-Qun Zhou	Aerial Image	YOLOv3	84%
Lianbing Deng	ship detection	YOLOv3	90%
Jingyi Zhao	Vision-Based Vehicle	YOLOv4	83%
Lizong Liu	Transmission Lines	YOLOv3	87%
Zhuang-Zhuang Wang	Small-Object	YOLO and Dense	83%
Zhonghua Hong	ship detection	YOLOv3	90%
Chengyuan Song	Strawberry Fruit Growth	YOLO	85%
Mansheng Xiao	Freight Train	BD-YOLO	90%
Jyun-Yu Jhang	Traffic-Monitoring	YOLO	88%
Xiaorong Xu	Infrared Image	YOLOv5	83%

Object detection has an additional purpose in addition to classifying picture objects; this additional duty is known as object localization and is the first significant complexity of object detection. Researchers most frequently employ a multi-task loss function to solve this problem by penalizing both classification and localization failures.

$$L(p, u, t^u, v) = L_c(p, u) + \lambda [u \geq 1] L_l(t^u, v) \quad \square \quad 1$$

The equation represents the classification (L_c) and localization (L_l).

It is possible to change the λ to give categorization or localization a higher priority.

The cons are listed below:

Illumination challenges: Rapid changes in lighting could result in the detection of false positive objects. For instance, indoor lighting may suddenly turn on or off, or the light source itself may shift. The shadows cast by moving objects, reflections off of shiny objects, and rapid transitions from bright sunlight to gloomy or rainy conditions are all possible outside. Also, there is always a chance that a moving object and the background will share the same hue. To prevent errors in the detection of moving objects, the backdrop

model should be able to adjust to variations in illumination and rapid changes in brightness.

Speed: Detectors must be trained to do analysis in a constantly changing environment when it comes to video. This means that to be able to recognize items that are moving, object detection algorithms must not only properly categorize relevant objects but also be extremely quick during prediction.

Unpredicted motion: It is difficult to recognize items moving suddenly. In particular, a vehicle's jackrabbit start might cause a tracker to lose track of the item or result in a tracking algorithm mistake. Objects that move either too slowly or too quickly are another source of detecting problems. The temporal differencing approach won't be able to identify an object's parts if it travels slowly. As an item moves quickly, a trail of ghost regions will appear after it in the foreground mask. Another difficulty is intermittent motion, which is when an item moves for a time, pauses and then resumes moving.

Occlusion: Moreover, occlusions can make it far more challenging to find and follow moving objects in videos. For example in the case, a car driving along the road could get buried by some tree branches or other things. Object tracking techniques are made more difficult by the possibility of objects in a video feed being completely or partially obscured.

III. PROPOSED WORK

The You Only Look Once (YOLO) is computer vision models-based family that includes the model known as YOLOv5. When it comes to object detection YOLOv5 plays the best role for recognizing different sizes, orientation, etc. s, m, l and x are the four major discrepancy of YOLOv5 which are named as Small, medium, large and extra-large respectively and each of them offers accuracy rates higher respectively. Each of the variations of YOLOv5 requires different amount of time to train.

Bring into existence a fresh technique after considering the flaws of earlier attempts. The approach proposed is primarily concerned with object detection in video sequences. Without using any applications from outside sources, the system can recognize things. Whether it is live input or pre-recorded, the video format is what counts the most. The system gathers information over a specified time period for the purpose of objection detection in a video sequence. The analysis is based on the discovered items, which are revealed in the output.

Implementation –

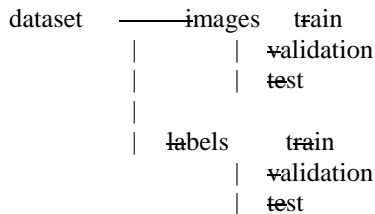
4.1 Object detection

The first step in the procedure is the collecting of images that have been carefully chosen for their superior quality, perception, and group or individual items (i.e. military weapons, missiles and drones).

Many websites (including roboflow, makesense.ai, and others) on the internet allow us to create a bounding box; roboflow, however, has some nice features like a bounding box tool, a smart polygon tool, and a polygon tool that aids in labeling the collected images for improved accuracy and validation. This clever polygon tool labels and encapsulates the things in the image. A data set is created once all the gathered images have been labeled.

Object detection features several state-of-the-art architectures that may be used on real-world datasets to recognize things with respectable accuracy. The sole requirement is that the test dataset has the same classes as the previously trained detector.

From the dataset for object detection in video sequence, some experiments are carried out. There are 1247 training images and 143 validation images with 30 class annotations. Depending upon the guidelines training set will undergo for training and validation set will undergo for evaluation of the performance.



An altered version of the YOLOv5 (Y_v) is created specifically for this work in order to evaluate and forecast image frames. When a video has L preceding frames, i.e., $t = 1, 2, \dots, L$, Y_v creates feature map F_t for each sequence.

The number neighboring frames that are associated with the current frames which have been aggregated is denoted by τ . While it is identical to other values of τ , we explain our shell in the case of $\tau = 1$ in the bellow sections for better understanding and convenience. The Y_v module generates the associated feature flow map, $M_{q \rightarrow p}^v$. Two pictures in tandem I_p and I_q , where I_p represents the picture at the current frame and I_q represents the picture at the frame next to it. Using the $M_{q \rightarrow p}^v$, The I_p feature map will get encased by I_q feature map. The encased feature map $F_{q \rightarrow p}$ is acquired as:

$$F_{q \rightarrow p} = \tau(F_q, M_{q \rightarrow p}^v)$$
, the bilinear encased function τ being used.

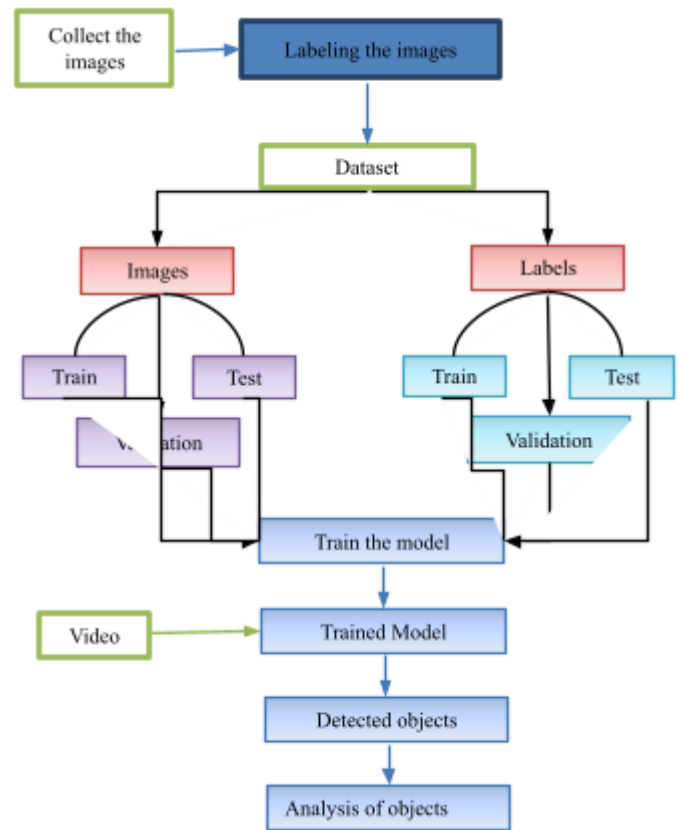


Fig.1 Flow diagram

Y_v employs fewer anchor boxes (split the input picture into an $N \times N$ grid). This was created with the help of darknet neural networks. Now, each frame will be subjected to gradient detection using $N \times N$ grids or cells. ResNet-50 consists of 50 layers. ResNet-50 is capable of loading over a million photos from a data collection. Floating point operations it has are 3.8×10^9 which can perform better operation. "x" is defined as identity.

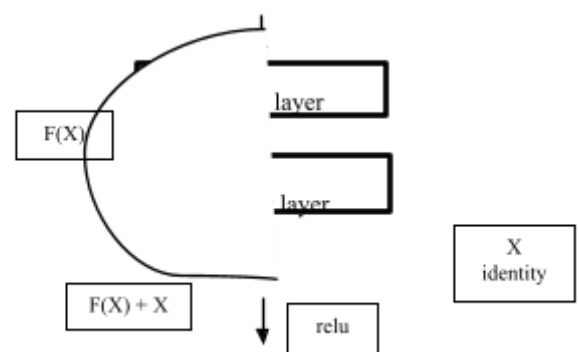


Fig.2 The identity will go through different weight layers with relu

The next step is to create an interface where we may arrange the structure of input and output. The website is made up of many buttons and sorts that allow the user to quickly browse. We may either feed the model live video input or

pre-recorded video input here. The model will process and analyze the video and present the detected items as a result.

4.2 Network architecture

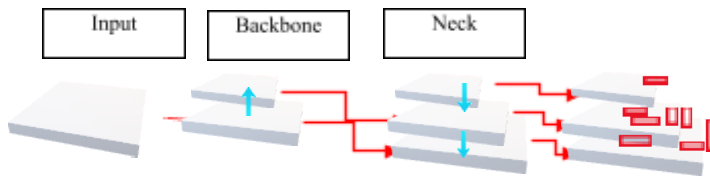


Fig.4 Network architecture

The YOLO is a member of the single-shot detector family. Network backbone: Input images will undergo extraction of essential characters which is done by Backbone. Valuable properties are extracted from an input image with the help of backbone called CSP (Cross Stage Partial Networks) in YOLOv5

4.3 Advantages

- More Reliable
- Data confidentiality
- For the calculation of each point, a sliding window effect is created.

4.4 Analysis

The same video is analyzed in the model throughout the video process, where it will study the items every frame. It examines the type of thing, displays the number of objects, and determines if it is dangerous or not. This study can be presented in a variety of formats, including text, graph, table, and histogram.

IV. RESULT AND DISCUSSION

The enhanced the performance of YOLO models by refining the feature map and including ResNet50 in the algorithm. The best model for object identification in the video was the YOLOv5 model, which had the best mAP of 87.8% and Frame Per Second of 108.4.

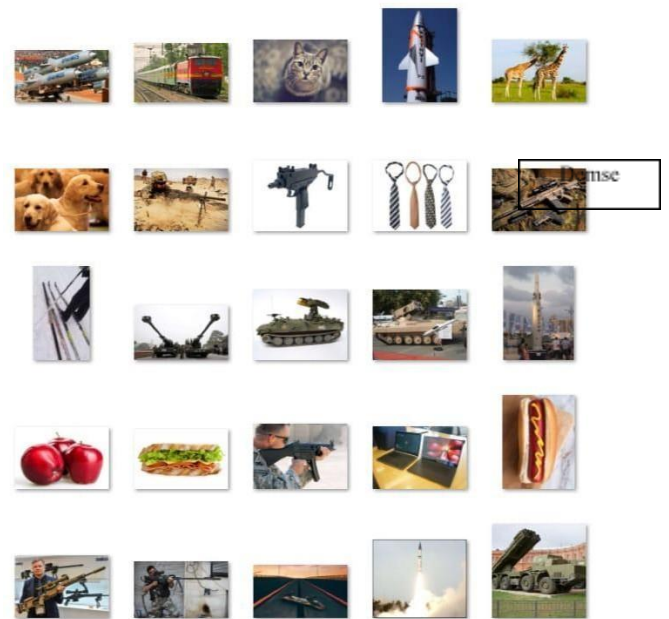


Fig.5 Sample Data

Sample data consists of all the collected images of military vehicle's, weapons, missiles, sniper rifles, dog, train etc.

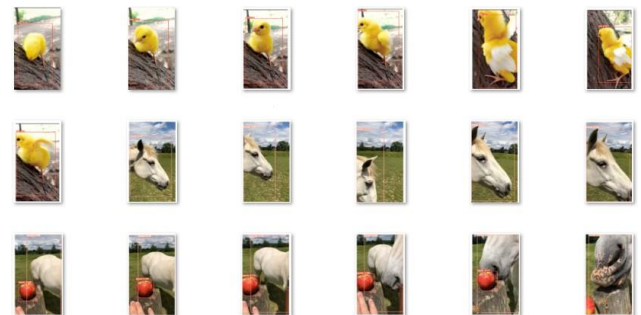


Fig.6 Results

The detected objects with the accuracy per frames are represented in sequence.

Table 2

Summarization about the Epoch with respect to loss functions and mAP50.

Epoch	box_loss	obj_loss	cls_loss	mAP50
0	0.1027	0.02558	0.04088	0.0472
1	0.07733	0.02893	0.03731	0.277
2	0.06347	0.0248	0.032	0.24
3	0.06	0.02445	0.02988	0.332
4	0.0595	0.02296	0.02726	0.298
5	0.05369	0.0209	0.02454	0.433
6	0.04928	0.0221	0.02267	0.248
7	0.04784	0.01951	0.02145	0.453
8	0.04583	0.0206	0.01986	0.551
9	0.04376	0.01904	0.01813	0.623

10	0.04186	0.01851	0.01838	0.635
11	0.03858	0.01864	0.0187	0.633
12	0.03864	0.01841	0.01841	0.583
13	0.03661	0.01819	0.01728	0.647
14	0.03695	0.01756	0.01731	0.654
15	0.03727	0.01866	0.01639	0.675
16	0.0344	0.01831	0.01577	0.667
17	0.03561	0.0178	0.01525	0.679
18	0.03557	0.01775	0.01576	0.738
19	0.03355	0.01748	0.01506	0.61
20	0.03379	0.01861	0.01506	0.657
21	0.03292	0.01794	0.016	0.691
22	0.03348	0.01802	0.01404	0.626
23	0.03396	0.01659	0.01496	0.719
24	0.0321	0.01769	0.01333	0.681
25	0.03216	0.01688	0.01338	0.62
26	0.03291	0.01657	0.01578	0.608
27	0.03287	0.01664	0.0136	0.65
28	0.03135	0.01748	0.01493	0.659
29	0.0323	0.01699	0.01387	0.597
30	0.0316	0.01759	0.01465	0.671
31	0.03247	0.01761	0.01135	0.762
32	0.03036	0.01686	0.01214	0.761
33	0.02997	0.01704	0.01094	0.723
34	0.02942	0.01576	0.01089	0.72
35	0.02992	0.01695	0.01288	0.698
36	0.03057	0.01607	0.01201	0.703
37	0.02749	0.01649	0.01326	0.73
38	0.02856	0.01582	0.01301	0.695
39	0.02898	0.01597	0.01223	0.733
40	0.02908	0.01628	0.01345	0.743
41	0.02804	0.01606	0.01359	0.745
42	0.02769	0.01667	0.01036	0.821
43	0.03043	0.01696	0.01111	0.681
44	0.02794	0.01598	0.01202	0.755
45	0.02694	0.01604	0.00886	0.741
46	0.02815	0.01573	0.01026	0.74
47	0.02902	0.01552	0.01234	0.675
48	0.02806	0.01549	0.01251	0.659
49	0.02756	0.01544	0.01131	0.769
50	0.02762	0.01613	0.01018	0.744
51	0.02644	0.01496	0.00973	0.73
52	0.02603	0.01445	0.01046	0.746
53	0.02754	0.01564	0.01195	0.763
54	0.02584	0.01496	0.01032	0.74
55	0.02507	0.01569	0.01122	0.768
56	0.02505	0.01524	0.012	0.775
57	0.02534	0.01469	0.00996	0.806
58	0.02633	0.01497	0.01152	0.826
59	0.02558	0.01486	0.01053	0.846
60	0.02565	0.0144	0.00922	0.878

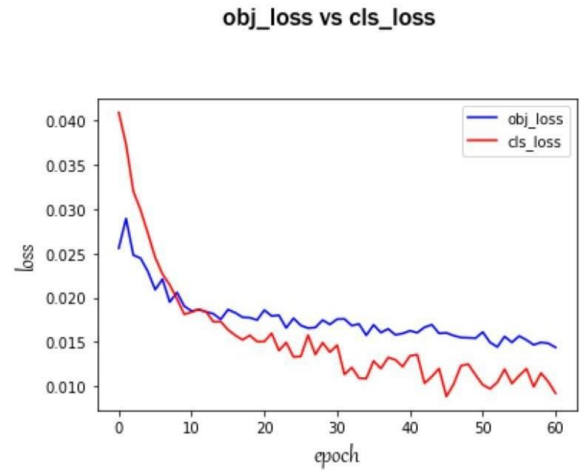


Fig.8 obj_loss vs cs_loss

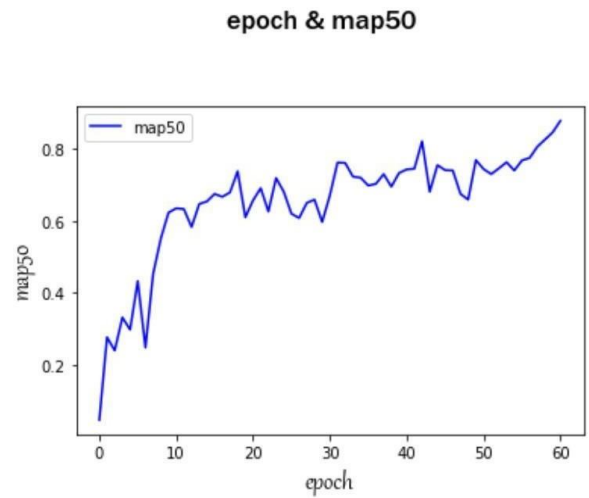


Fig.9 epoch & map50

V. CONCLUSION AND FUTURE WORK

The put forth an enhanced target recognition approach for video sequences based on YOLOv5. In the future, additional types of targets can be added, including armor, helicopters, sheep, pizza, and oranges. By enhancing the dataset, we further confirm the suggested model's detection ability. A self-built dataset will be used to validate the enhanced method's detection effectiveness, and it is applied to YOLOv4 and YOLOv3.

VI. REFERENCES

- [1] Isa, I. S., Rosli, M. S. A., Yusof, U. K., Maruzuki, M. I. F., & Sulaiman, S. N. (2022). Optimizing The Hyperparameter Tuning of YOLOv5 For Underwater Detection. *IEEE Access*.
- [2] Luo, Shan, and Jihong Liu. "Research on Car License Plate Recognition Based on Improved YOLOv5m and LPRNet." *IEEE Access* 10 (2022): 93692-93700.
- [3] Wang, Chenglong, Ziran Zhou, and Zhiming Chen. "An Enhanced YOLOv4 Model with Self-Dependent Attentive Fusion and Component Randomized Mosaic Augmentation

for Metal Surface Defect Detection." *IEEE Access* 10 (2022): 97758-97766.

[4] Kong, Lingren, Jianzhong Wang, and Peng Zhao. "YOLO-G: A Lightweight Network Model for Improving the Performance of Military Targets Detection." *IEEE Access* (2022).

[5] Yang, Kuihe, and Ziyang Song. "Deep Learning-Based Object Detection Improvement for Fine-Grained Birds." *IEEE Access* 9 (2021): 67901-67915.

[6] Zhou, Li-Qun, Peng Sun, and Jin-Chun Piao. "A Novel Object Detection Method in City Aerial Image Based on Deformable Convolutional Networks." *IEEE Access* 10 (2022): 31455-31465.

[7] Li, H., Deng, L., Yang, C., Liu, J., & Gu, Z. (2021). Enhanced YOLO v3 tiny network for real-time ship detection from visual image. *Ieee Access*, 9, 16692-16706.

[8] Zhao, J., Hao, S., Dai, C., Zhang, H., Zhao, L., Ji, Z., & Ganchev, I. (2022). Improved Vision-Based Vehicle Detection and Classification by Optimized YOLOv4. *IEEE Access*, 10, 8590-8603.

[9] Li, Hui, Lizong Liu, Jun Du, Fan Jiang, Fei Guo, Qilong Hu, and Lin Fan. "An Improved YOLOv3 for Foreign Objects Detection of Transmission Lines." *IEEE Access* 10 (2022): 45620-45628.

[10] Wang, Zhuang-Zhuang, Kai Xie, Xin-Yu Zhang, Hua-Quan Chen, Chang Wen, and Jian-Biao He. "Small-Object Detection Based on YOLO and Dense Block via Image Super-Resolution." *IEEE Access* 9 (2021): 56416-56429.

[11] Hong, Zhonghua, Ting Yang, Xiaohua Tong, Yun Zhang, Shenlu Jiang, Ruyan Zhou, Yanling Han, Jing Wang, Shuhu Yang, and Sichong Liu. "Multi-scale ship detection from SAR and optical imagery via a more accurate YOLOv3." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021): 6083-6101.

[12] An, Qilin, Kai Wang, Zhongyang Li, Chengyuan Song, Xiuying Tang, and Jian Song. "Real-Time Monitoring Method of Strawberry Fruit Growth State Based on YOLO Improved Model." *IEEE Access* 10 (2022): 124363-124372.

[13] Zhang, L., Wang, M., Liu, K., Xiao, M., Wen, Z., & Man, J. (2022). An Automatic Fault Detection Method of Freight Train Images Based on BD-YOLO. *IEEE Access*, 10, 39613-39626.

[14] Lin, Cheng-Jian, and Jyun-Yu Jhang. "Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks." *IEEE Access* 10 (2022): 14120-14133.

[15] Li, S., Li, Y., Li, Y., Li, M., & Xu, X. (2021). YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. *IEEE Access*, 9, 141861-141875.