Title:Object Detection using Deep Learning framework YOLO”

A Report Submitted in Partial Fulfillment of the Requirements

for the

6th Semester B.Tech. Summer Internship

(Mode of Internship is OFFLINE)

By

Your Name	**Scholar ID**
Satyam Prasad Deva Sarmah	1912115
Arpita Bhattacharjee	1912003

Under the Guidance of

Dr. Ripon Patgiri

Assistant Professor

Computer Science and Engineering Department

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Assam

1st June-15th July 2022

DECLARATION

“ Title : Detection of object using Deep Learning framework YOLO”

We declare that the presented work represents largely our own ideas and work in our own words. Where others ideas or words have been included, we have adequately cited and listed in the reference materials. We have adhered to all principles of academic honesty and integrity.

Sl.No.	Scholar ID	Scanned Signature
1.	1912115	Satyam Prasad Deva Sarmah
2.	1912003	Arpita Bhattacharjee

Department of Computer Science and Engineering
National Institute Of Technology Silchar, Assam

ABSTRACT

With the recent development of technology, object detection technology is emerging, and these technologies can also be applied to illegal immigrants, industrial and natural disasters, and missing people and objects. In this project, we would like to explore ways to increase object detection performance in these situations. Object detection techniques are the foundation for the artificial intelligence field. This research report gives a brief overview of the You Only Look Once (YOLO) algorithm and its working in its small version. Through the analysis, we reach many remarks and insightful results. In this, we're proposing a solution to detect mainly the air vehicle,other objects and the numbers that each angle provides. There are already some methods proposed that use deep learning for recognizing drone-type air vehicles. However, the effective uses of these models are limited. We are proposing a YOLOv5-based solution as it's lightweight, fast, and has good accuracy.

LIST OF FIGURES

Figure

Contents

1. INTRODUCTION

1.1 Motivation

Modern commercial drones have several characteristics that make them an aerial threat to facilities such as airports, stadiums, oil and gas refineries, and prisons. A drone flying near a runway endangers planes during takeoff and landing. Because of their ability to bypass ground security, drones can also pose a threat to stadiums, prisons, and oil and gas facilities. As a result, security teams require a method to detect drones in the air and keep track of what is flying in their airspace. There are numerous technologies and solutions available to assist security teams in tracking drones. Drone detection technology is used to detect unmanned aircraft systems (UAS), also known as drones. There are different technologies used to detect drone activity in the airspace.

You Only Look Once (YOLO) is a popular and well-known algorithm. YOLO is well-known for its ability to detect objects. The first YOLO version was introduced in 2015 by Redmon et al. Scholars have published several YOLO subsequent versions over the years, dubbed YOLO V2, YOLO V3, YOLO V4, and YOLO V5. There are a few revised-limited versions, such as YOLO-LITE. This paper discusses how to detect well in a confusing environment to recognize objects in order to solve these problems. We were able to significantly improve the model's performance by modifying the Conv layer, the main layer of the original YOLOv5.

1.2 Problem Statement

We are trying to detect relevant objects from any image using YOLOv5.YOLOv5 has some performance improvements, with the following significant advantages:

• The PyTorch framework is user-friendly and easy to train your data set, making it easier to put into production than the Darknet framework used in YOLO V4;

• Easy to read code, integration of a large number of computer vision technology, is conducive to learning and reference;

• Easy to configure the environment, model training is very fast, and batch reasoning produces real-time results.

YOLO V5 provides each batch of training data through the data loader and enhances the training data simultaneously. The data loader performs three types of data enhancement: scaling, color space adjustment, and mosaic enhancement. The data proves that Mosaic enhancement can indeed effectively solve the most troublesome small object problem in the model training. That is, the small object detected is not as accurate as of the large object.YOLO V5 can flexibly control models from 10+M to 200+M, and its small model is very impressive.

1.3 Objective

Our objective was to detect air vehicle drone using deep learning framework YOLO V5.

Most reliable technologies for drone detection-Drone Detection using Radio Frequency Technology,Drone Detection using Rader, Drone Detection using Visual Tracking

<table>
<tr><th colspan="6">Object Detection</th></tr>
<tr><th colspan="3">Detection Components</th><th>Learning Strategy</th><th colspan="2">Applications & Benchmarks</th></tr>
<tr><th>Detection Settings</th><th>Detection Paradigms</th><th>Backbone Architecture</th><th>Training Stage</th><th colspan="2">Applications</th></tr>
<tr><td rowspan="2">Bounding Box</td><td rowspan="2">Two-Stage Detectors</td><td>VGG16,ResNet,DenseNet</td><td>Data Augmentation</td><td rowspan="2">Face Detection</td></tr>
<tr><td rowspan="2">MobileNet, ResNeXt</td><td>Imbalance Sampling</td></tr>
<tr><td rowspan="2">Pixel Mask</td><td rowspan="2">One-Stage Detectors</td><td>Localization Refinement</td><td rowspan="2">Pedestrian Detection</td></tr>
<tr><td>DetNet, Hourglass Net</td><td>Cascade Learning</td></tr>
<tr><td colspan="2"></td><td></td><td>Others</td><td>Others</td></tr>
<tr><th colspan="2">Proposal Generation</th><th>Feature Representation</th><th>Testing Stage</th><th colspan="2">Public Benchmarks</th></tr>
<tr><td colspan="2">Traditional Computer Vision Methods</td><td>Multi-scale Feature Learning</td><td rowspan="2">Duplicate Removal</td><td rowspan="2">MSCOCO, Pascal VOC, Open Images</td></tr>
<tr><td colspan="2">Anchor-based Methods</td><td>Region Feature Encoding</td></tr>
<tr><td colspan="2">Keypoint-based Methods</td><td>Contextual Reasoning</td><td rowspan="2">Model Acceleration</td><td>FDDB, WIDER FACE</td></tr>
<tr><td colspan="2">Other Methods</td><td>Deformable Feature Learning</td><td>Others</td><td>KITTI, ETH, CityPersons</td></tr>
</table>

Fig:1 We categorize various contributions for deep learning based object detection into three major categories: Detection Components, Learning Strategies, Applications and Benchmarks. We review each of these categories in detail.

YOLO is developed based on Convolutional Neural Network (CNN) and can produce fast and effective object detection. In the YOLO(You Only Look Once) method, the input images are only seen once through the neural network and it predicts the detected object in the image. It works by dividing the input image into different grids based on predefined grid size and then predicts the probability of the desired object in each grid. It predicts all the classes and the object bounding that are in the image in one run of the Algorithm. Thus it became a very fast end-to-end object detection and can be used for real-time detection. There are also continuous improvements on the YOLO algorithm in terms of accuracy, speed, and lightweight.

2. Methodology

A. Data Collection and Pre-Processing

1) Dataset: This dataset is prepared for our 2019 year "Amateur Drone Detection and Tracking" project. There are more than 4000 amateur drone pictures in the dataset, which is usually trained with amateur (like dji phantom) drones. In addition, the dataset contains non-drone, drone-like "negative" objects.

2)Dataset Split: The data set is split into 3 parts. 80% data was for the test set, 10% for validation and 10% for test. The original data set contains 4000 images. We apply augmentation in the test set and each image has 3 augmented versions.

3)data=data\coco128.yaml, imgsz=[640, 640], conf_thres=0.25, iou_thres=0.45, max_det=1000, device=, view_img=False, save_txt=False, save_conf=False, save_crop=False, nosave=False, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, project=runs\detect, name=exp, exist_ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False

4)The NVIDIA® CUDA® Toolkit provides a development environment for creating high-performance GPU-accelerated applications. With the CUDA Toolkit, we developed, optimized, and deploy your applications on GPU-accelerated embedded systems, desktop workstations, enterprise data centers, cloud-based platforms and HPC supercomputers. The toolkit includes GPU-accelerated libraries, debugging and optimization tools, a C/C++ compiler, and a runtime library to deploy your application.Using built-in capabilities for distributing computations across multi-GPU configurations, scientists and researchers can develop applications that scale from single GPU workstations to cloud installations with thousands of GPUs.

B. Deep Learning Architecture

YOLOv5 is the deep learning-based architecture that is used to conduct this experiment. YOLOv5 is lightweight and fast and also needs less computational power than the other current state-of-the-art architecture model while keeping the accuracy near to the current state-of-the-art detection models. It is much faster than the other YOLO models. YOLOv5 uses CSPNET as the backbone to extract the feature map from the image. It also uses Path Aggregation Network (PANet) to boost the information flow. The following image shows the architecture of YOLOv5. We are using YOLOv5 for the following reasons:

1) Has useful components such as state-of-the-art activation function, hyperparameter, data augmentation technique and a convenient manual

2) Its lightweight architecture makes it computationally easy to train with small resources.

3) The size of the model is quite small and lightweight, thus can be used with mobile devices.

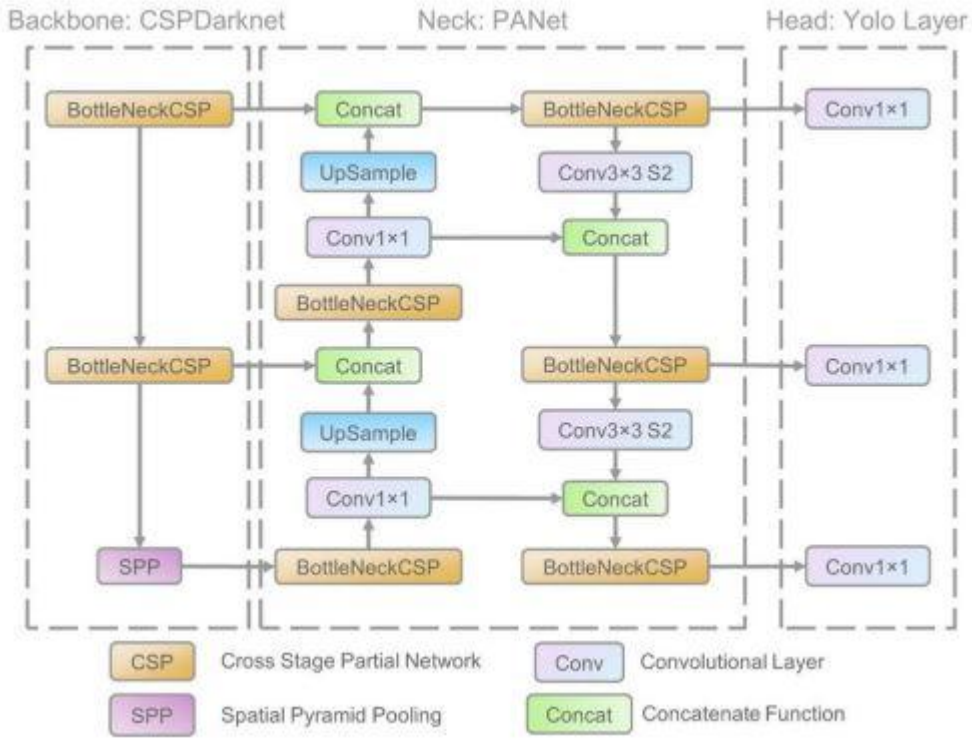

Fig. 2. YOLOv5 network architecture

C. Model Training Process

The approach is to fine-tune the YOLOv5x model which is already pre-trained on the COCO data set. weights=None, # model.pt path(s); batch_size=32, # batch size; conf_thres=0.001, # confidence threshold iou_thres=0.6, # NMS IoU thresholddevice='', # cuda device, i.e. 0 or 0,1,2,3 or cpu;single_cls=False, # treat as single-class dataset; augment=False, # augmented inference;verbose=False, # verbose output;save_txt=False, # save results to *; save_hybrid=False, # save label+prediction hybrid results to *.txt;save_conf=False, # save confidences in --save-txt labels; save_json=False, # save a COCO-JSON results file;project=ROOT / 'runs/val', # save to project/name;name='exp', # save to project/name; exist_ok=False, # existing project/name ok, do not increment; half=True, # use FP16 half-precision inference; dnn=False, # use OpenCV DNN for ONNX inference;model=None; dataloader=None;save_dir=Path('');plots=True; callbacks=Callbacks();compute_loss=None,

3)Experimental Result

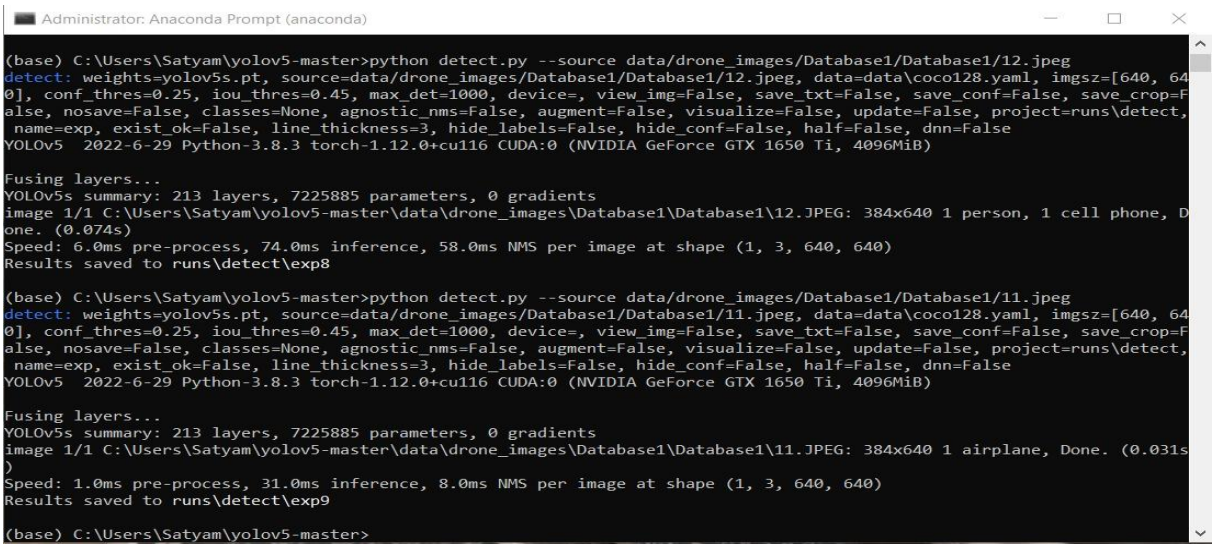

Fig 3 : Results

4.1)Conclusion

The experiment is based on the Yolo v5s architecture. Yolo v5s is a pretrained model the second smallest and the fastest model available. Here we have experimented on detecting drones from a unified , real time dataset of more than 4000 images. The Yolo v5s framework trains on l on COCO128 by specifying dataset, batch-size, image size and either pretrained, or is randomly initialized . For visualization purpose ClearML is used which is completely integrated into YOLOv5 framework to track our experimentation, manage dataset versions and even remotely execute training runs. Weights and biases is integrated with YOLOv5 for real-time visualization and cloud logging of training runs. Here the framework have identified the drones with good accuracy and the accuracy is shown in terms of confidence threshold. The F1 scores also shows that Yolo is very good in detecting unifie, real time objects. CUDA also helped the model to identify the objects at great pace. The COCO128 has 128 parameters fitted into its model from which any object on the images can be identified. But there is still room for improvement . Future research on creating different version can bring even more accuracy to the model by adding more and more parameters into it.

4.2)Reference

1)A Review of Yolo Algorithm Developments A Review of Yolo Algorithm Developments

Peiyuan Jiang, Daji Ergu*, Fangyao Liu, Ying Cai, Bo Ma

2)Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions Hyun-Ki Jung * and Gi-Sang Choi

3)YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles

Aduen Benjumea* Izzeddin Teeti† Fabio Cuzzolin† Andrew Bradley*

4)Using YOLOv5 Algorithm to Detect and Recognize American Sign Language

Tasnim Ferdous Dima,MD. Eleas Ahmed

5)Certificates

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

(AN INSTITUTE OF NATIONAL IMPORTANCE UNDER NIT ACT-2007)

Silchar, Cachar, Assam-788010

"Satyendra Nath Bose Summer Internship Program 2022"

Certificate of Participation

This is to certify that Miss./Mr. Satyam Prasad Deva Sarmah

from National Institute of Technology, Silchar *has successfully completed the Summer Internship from* 1st June 2022 to 15th July 2022 *at NIT Silchar under mentorship of Dr./Prof.* Dr. Ripon Patgiri *in* CSE *Department.*

Date: 14/7/22
Place: UGRC Coordinator

Mentor

(Prof. Sivaji Bandyopadhyay)
Director, NIT Silchar

NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

(AN INSTITUTE OF NATIONAL IMPORTANCE UNDER NIT ACT-2007)

Silchar, Cachar, Assam-788010

"Satyendra Nath Bose Summer Internship Program 2022"

Certificate of Participation

This is to certify that Miss./Mr. Arpita Bhattacharjee

from National Institute of Technology, Silchar *has successfully completed the Summer Internship from* 1st June 2022 to 15th July 2022 *at NIT Silchar under mentorship of Dr./Prof.* Ripon Patgiri *in* Computer Science & Engineering *Department.*

Date: 14/7/22
Place: UGRC Coordinator

Mentor

(Prof. Sivaji Bandyopadhyay)
Director, NIT Silchar