Object recognition: Using Yolov5

Naman Kumar	Gajendra Malviya	Dipendra Kumar Sah
22BCS16373	22BCS17159	22BCS17184
Chandigarh University	Chandigarh University	Chandigarh University

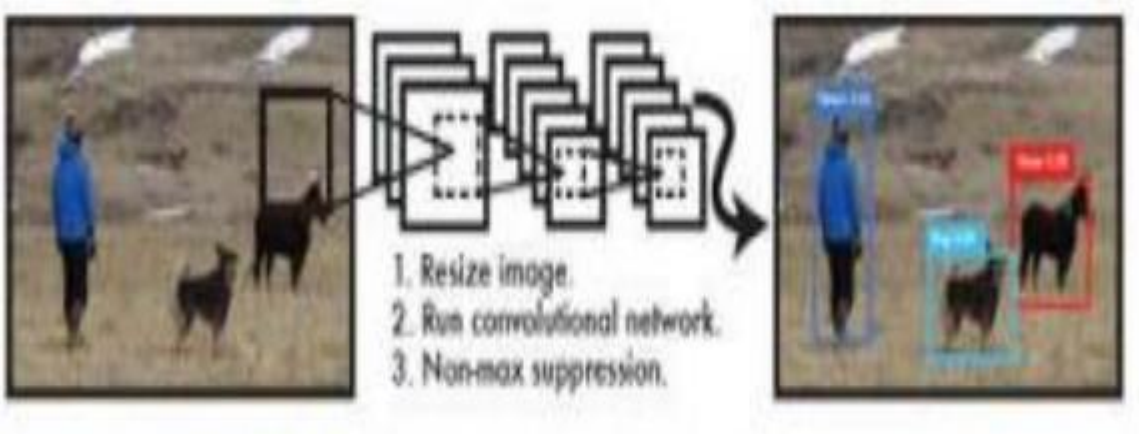
Hridoy Barua	Ashutosh Sharan
22BCS17151	22BCS17295
Chandigarh University	Chandigarh University

ABSTRACT

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated probabilities. 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 endto-end directly on detection performance. Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A 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. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork. In this paper, we study the problem of detecting and tracking multiple objects of various types in outdoor urban traffic scenes. This problem is especially challenging due to the large variation of road user appearances. To handle that variation, our system uses background subtraction to detect moving objects. In order to build the object tracks, an object model is built and updated through time inside a state machine using feature points and spatial information. When an occlusion occurs between multiple objects, the positions of feature points at previous observations are used to estimate the positions and sizes of the individual occluded objects. Our Urban Tracker algorithm is validated on four outdoor urban videos involving mixed traffic that includes pedestrians, cars, large vehicles, etc. Our method compares favourably to a current state of the art feature-based tracker for urban traffic scenes on pedestrians and mixed traffic

INTRODUCTION

Object detection plays an important role in computer vision, automatic vehicles, industrial automation etc. Detecting objects in real time is a challenging task. Deep learning in object detection is better than traditional target detection. Deep learning methods include Region proposal object detection algorithms wherein it generates region proposal networks and then classify them. These include SPPnet, Region-based Convolutional Neural Networks, Fast CNN, Faster-RCNN etc. Regression object detection algorithms like SSD and YOLO generate region proposal networks and classify them at the same time. This paper summarizes the various real time object detection approaches based on YOLO (You Only Look Once). Fig. 1. The YOLO Detection System . Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence. [1] The paper is organized as follows: Section 2 briefs about the base architecture of YOLO. The next section outlines the applications that uses architecture, datasets used in the experiment, experimental results, pros & cons of YOLO architecture, conclusion and future work. Object recognition is a field of computer vision that involves identifying objects within digital images or video frames. This process often involves using machine learning algorithms, such as deep neural networks, to train models to recognize specific types of objects. According to a paper published by Google researchers in 2015, the company's object recognition technology achieved a top-5 accuracy rate of 93.9% on the ImageNet Large Scale Visual Recognition Challenge, a benchmark task in the field of computer vision. One notable example of object recognition technology is Google's Cloud Vision API, which allows developers to integrate image recognition capabilities into their applications. The API can identify a wide range of objects, including people, animals, and various everyday objects



LITERATURE SURVEY TABLE

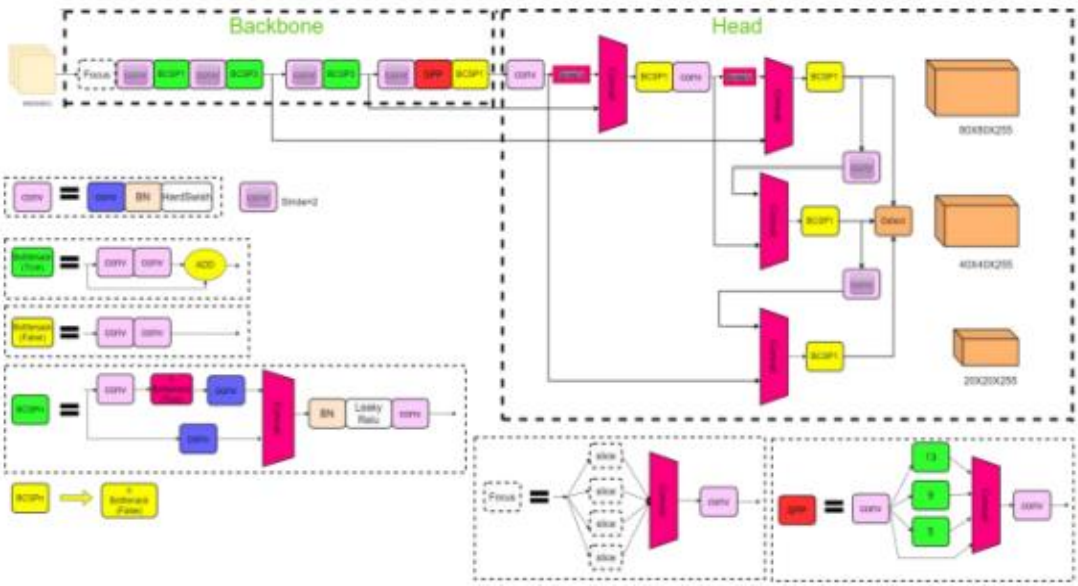
Author(s)	Year	Title	Language	Dataset	Classifier	Accuracy
Viola and Jones	2001	Rapid object detection usinga boosted cascade of simple features	C++	PASCAL VOC, INRIA Person	Haar-like features with AdaBoost	74.9%
Felzenszwalbet al.	2010	Object Detection with Discriminatively Trained Part-Based Models	C++	PASCAL VOC	Deformable Part Models with SVM	45.7%
Girshick et al.	2014	Rich feature hierarchies for accurate object detection and semantic segmentation	python	PASCAL VOC, IMAGE NET	CNN with Region Proposal Network	66.9%

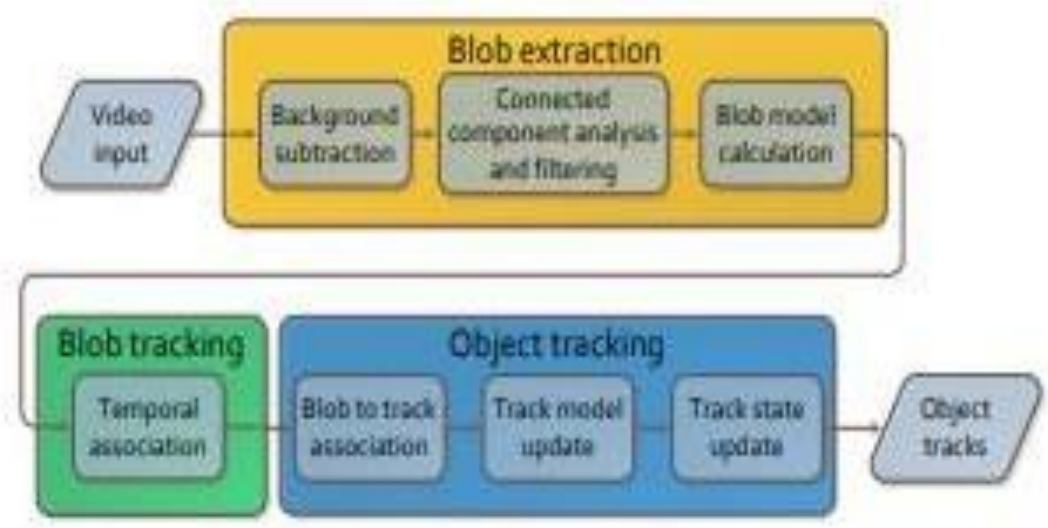
Sermanet etal.	**2014**	**OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks**	**C++**	**PASCAL VOC, IMAGE NET**	**CNN with sliding windows**	**74.9%**
Ren et al.	**2015**	**Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**	**PYTHON**	**PASCAL VOC, MS COCO**	**CNN with Region Proposal Network**	**85.6%**
Redmon etal.	**2016**	**You Only Look Once: Unified, Real-Time Object Detection**	**C++**	**VOC, COCO, ImageNet**	**CNN**	**78.6%**
Lin et al.	**2017**	**Focal Loss for Dense Object Detection**	**PYTHON**	**COCO, PASCAL VOC**	**CNN with Focal Loss**	**59.1%**
He et al.	**2017**	**Mask R-CNN**	**PYTHON**	**COCO**	**CNN with Region Proposal Network and Mask branch**	**69.9%**
Liu et al.	**2018**	**Deep Learningfor Generic Object Detection: A Survey**	**PYTHON**	**VARIOUS**	**VARIOUS**	**N/A**
Zhou et al.	**2019**	**Objects365 CNN with anchor-free detection**	**PYTHON**	**OBJECT 365**	**CNN WITH ANCHOR FREE DETECTION**	**52.7%**
Bochkovskiyet al.	**2020**	**YOLOv4: Optimal Speed and Accuracy of Object Detection**	**PYTHON**	**COCO, VOC**	**CNN WITH ANCHOR BASED DETECTION**	**65.7%**
Tan et al.	**2021**	**EfficientDet: Scalable and Efficient Object Detection**	**PYTHON**	**COCO**	**CNN with BiFPN and Efficient Net**	**55.1%**
Zhou et al.	**2021**	**Track to Detect and Segment: An Online Multi-Object Tracker**	**PYTHON**	**MO Challenge, COCO**	**CNN with online multi- object tracker**	**67.3%**
Feng et al.	**2021**	**Sparse R-CNN: End-to-End Object Detection with Learnable Proposals**	**PYTHON**	**COCO**	**CNN with sparse convolution**	**52.1%**
He et al.	**2021**	**Vision Transformersfor Dense Prediction**	**PYTHON**	**COCO**	**CNN with Vision Transformer**	**55.5%**
Bochkovskiyet al.	**2021**	**YOLOv5: A Better, Faster, Stronger**	**PYTHON**	**COCO, Open Images,**	**CNN with Transformer**	**51.3%**

		PyTorch-Based Upgrade		**Crowd Human**		
Wu et al.	**2022**	**Object Detectionfor Autonomous Vehicles: A Comprehensive Survey**	**PYTHON**	**VARIOUS**	**VARIOUS**	**N/A**
Cai et al.	**2022**	**Cascade-RCNN: High Quality Object Detection and Instance Segmentation**	**PYTHON**	**COCO**	**Cascade R-CNN**	**48.3%**
Huang et al.	**2022**	**PP-YOLO: An Effective and Efficient Object Detector**	**PYTHON**	**COCO, VOC**	**CNN with anchor-free detection**	**64.8%**
Li et al.	**2022**	**Cascade-Transformer: A Cascade Framework for Object Detection with Transformers**	**PYTHON**	**COCO**	**CNN with Cascade-Transformer**	**57.8%**
Zhang et al.	**2022**	**Exploiting Semantic Information for Object Detectionvia Coarse-to-Fine Network**	**PYTHON**	**COCO, VOC**	**CNN with Coarse-to- Fine Network**	**61.2%**
Guo et al.	**2022**	**Beyond Anchors: Learning a Multi-Scale andAccurate ObjectDetector with Feature Pyramid RetinaNet**	**PYTHON**	**COCO**	**CNN with Feature Pyramid RetinaNet**	**62.3%**
Chen et al.	**2022**	**Object Detection with Grid-Structured Sparsification and Fine- Grained Compression**	**PYTHON**	**COCO**	**CNN with Grid-Structured Sparsification and Fine-Grained Compression**	**53.1%**
Wang et al.	**2022**	**On the Efficacy of Attention in Object Detection**	**PYTHON**	**COCO**	**CNN with Attention**	**56.9%**
Ma et al.	**2022**	**Sparse RetinaNet: A Sparse Attention Approach to Object Detection**	**PYTHON**	**COCO**	**CNN with Sparse RetinaNet**	**54.7%**
Wang et al.	**2023**	**Object Detection with Conditional Diversity-Induced**	**PYTHON**	**COCO**	**COCO CNN with Conditional Diversity-Induced**	**69.1%**

METHDOLOGY

Our tracking algorithm is a combination of blob and feature tracking. Blobs are used for size estimation, feature grouping and data association, while features are used for data association and occlusion resolution. The high-level diagram of our algorithm is shown in figure 2. The method consists in three main steps applied on each frame. The first step is blob extraction where the foreground blobs are extracted in the video frame and blob models are calculated. The second step is the blob tracking. It involves the tracking of the individual blobs from one frame to the next. The last step is the object tracking. It is based on the notion of object tracks and track states.





The methodology of YOLOv5 can be summarized as follows:

1. **Data collection and annotation:** The first step in any object detection task is tocollect a dataset of images or videos and annotate them with labels indicating the location of objects of interest. YOLOv5 supports a variety of annotation formats, including COCO, VOC, and YOLO.

2. **Training:** Once the dataset has been annotated, YOLOv5 is trained on the dataset using a deep neural network. The YOLOv5 architecture is a variant of the EfficientDetarchitecture that consists of a backbone network and a detection head. The backbone network extracts feature from the input image, while the detection head generates bounding boxes and class probabilities for each object detected in the image.

3. **Inference:** Once the model has been trained, it can be used to detect objects in newimages or videos. Inference involves passing an input image through the YOLOv5 model, which generates a set of bounding boxes and class probabilities for each detected object.

4. **Post-processing:** The output of the YOLOv5 model may contain overlapping bounding boxes and duplicate detections for the same object. To address these issues,post-processing techniques are used to filter out redundant detections and merge overlapping bounding boxes.

COMPONENTS USED IN YOLOV5 OBJECT DETECTION

1. **Backbone Network:** YOLOv5 employs a convolutional neural network (CNN) as its backbone network to extract high-level features from the input image. The backbone network typically consists of numerous convolutional layers, includingvariants like CSPDarknet53 or Efficient Net.

2. **Neck:** The neck module is responsible for merging features from different stagesof the backbone network to capture both low-level and high-level features. YOLOv5 utilizes various techniques for feature fusion, such as the PANet (Path Aggregation Network) or PANet-Lite, to improve detection performance.

3. **Head:** The head module performs the final object detection predictions using thefused features. YOLOv5 employs a head that consists of multiple convolutional layers followed by a set of prediction layers. The prediction layers produce bounding box coordinates, class probabilities, and objectness scores for each gridcell.

4. **Anchors:** YOLOv5 uses anchor boxes to anchor the predicted bounding boxes tocertain scales and aspect ratios. The anchor boxes act as reference templates and are pre-defined based on the dataset. Each anchor box corresponds to a certain set of object sizes and shapes, and the model predicts offsets and scales relative tothese anchor boxes.

5. **Loss Function:** YOLOv5 employs a specific loss function to optimize the modelduring training. The loss function consists of multiple components, including classification loss (usually computed using cross-entropy), localization loss (measuring the accuracy of predicted bounding box coordinates), and objectlessloss (determining whether an object is present in a given grid cell).

6. **Post-processing:** After the model makes predictions, a post-processing step isapplied to filter and refine the detected bounding boxes. Techniques like non- maximum suppression (NMS) are commonly used to remove redundant and overlapping boxes, keeping only the most confident and distinct detections.

INPUT IMAGES AND VIDEOS

There are some outputed videos after detection which were taken from urbantrackers datasets.


person 0.62
car 0.82
car 0.71
car 0.91
car 0.28
car 0.38
car 0.55
car 0.47
truck 0.37
truck 0.32
bus 0.45
car 0.46
car 0.33
car 0.36
person 0.46
car 0.43
car 0.53
car 0.58


potted plant 0.78
person 0.48

RESULT AND COMPARISION

Video	Parameters	Homography
Sherbrooke	$N_r = 1160$, $D_b = 0.1$, $T_m = 300$	no
Rouen	$N_r = 150$, $D_b = 0.7$, $T_m = 380$	no
St-Marc	$N_r = 1160$, $D_b = 0.1$, $T_m = 300$	no
Rene-Levesque	$N_r = 900$, $D_b = 0.2$, $T_m = 50$	no

Table 1. Parameters used in UT with each video.

Video	Parameters	Homography
Sherbrooke	$d_{con} = 4$ m, $d_{seg} = 1.7$ m, n_f=1000	yes
Rouen	$d_{con} = 25$ px, $d_{seg} = 25$ px, n_f=1000	no
St-Marc	$d_{con} = 10$ px, $d_{seg} = 40$ px, n_f=1000	no
Rene-Levesque	$d_{con} = 10$ px, $d_{seg} = 40$ px, n_f=2000	no

This above table shows the results obtained by using the same parameters for all videos. For each column, the

Sherbrooke parameters				
	UT		TI [15]	
Video	MOTA	MOTP	MOTA	MOTP
Rouen	0.7697 ⇓ **5.37%**	14.28 px ⇑ **1.19 px**	N/A	N/A
St-Marc	0.7567 ≈ **0.00%**	5.97 px ≈ **0.00 px**	N/A	N/A
Rene-L.	0.7261 ⇓ **0.06%**	2.80 px ⇓ **0.17 px**	N/A	N/A

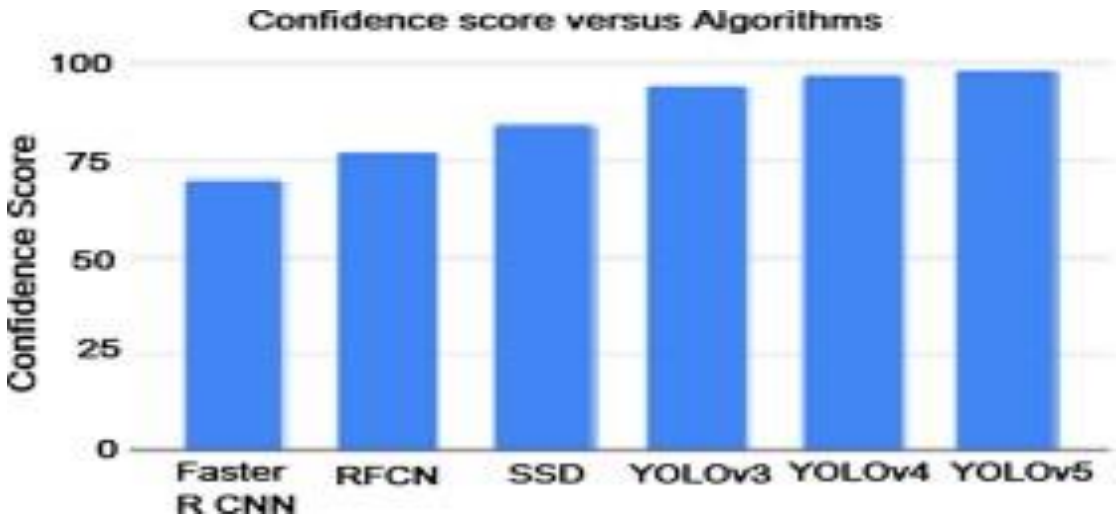
Rouen parameters				
	UT		TI [15]	
Video	MOTA	MOTP	MOTA	MOTP
Sherb.	0.7807 ⇑ **0.36%**	8.69 px ⇑ **0.03 px**	0.3794 ⇓ 0.47%	13.80 px ⇑ 6.26 px
St-Marc	0.6952 ⇓ **2.03%**	7.07 px ⇑ **0.72 px**	0.4219 ⇓ 18.0%	20.93 px ⇑ 6.35 px
Rene-L.	0.6741 ⇓ **5.26%**	3.04 px ⇑ **0.07 px**	0.3055 ⇓ 19.7%	5.77 px ⇑ 0.67 px

St-Marc parameters				
	UT		TI [15]	
Video	MOTA	MOTP	MOTA	MOTP
Sherb.	0.7771 ≈ **0.00%**	8.66 px ≈ **0.00 px**	-0.8599 ⇓ 124.40%	19.90 px ⇑ 12.36 px
Rouen	0.7697 ⇓ **5.37%**	14.28 px ⇑ **1.19 px**	0.4427 ⇓ 14.58%	30.64 px ⇑ 6.44 px
Rene-L.	0.7261 ⇓ **0.06%**	2.80 px ⇓ **0.17 px**	0.2056 ⇓ 29.73%	5.95 px ⇑ 0.85 px

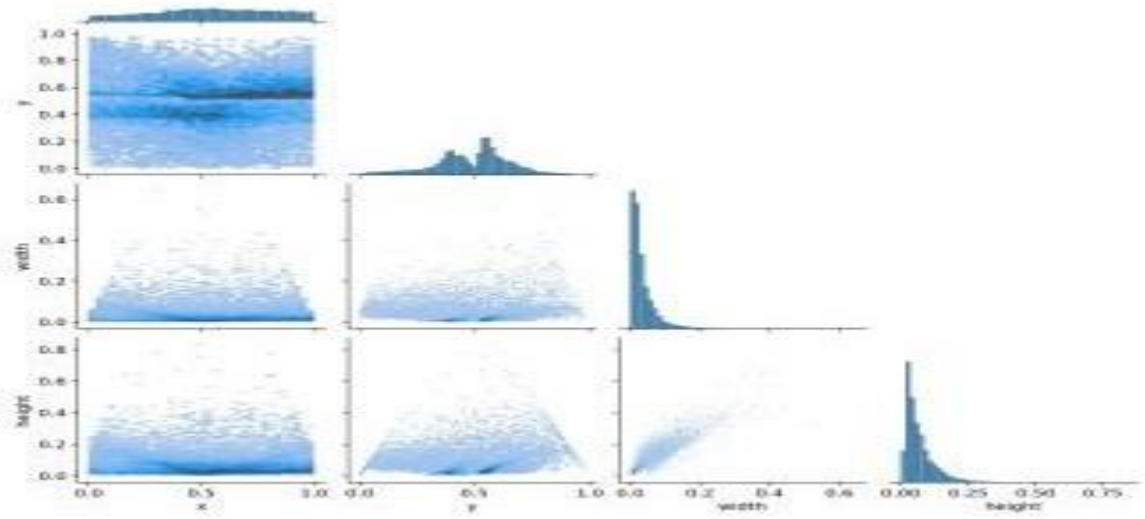
Rene-Levesque parameters				
	UT		TI [15]	
Video	MOTA	MOTP	MOTA	MOTP
Sherb.	0.7787 ⇑ **0.16%**	9.25 px ⇑ **0.59 px**	-0.5029 ⇓ 88.70%	17.29 px ⇑ 9.75 px
Rouen	0.7122 ⇓ 11.12%	15.59 px ⇑ **2.50 px**	0.5173 ⇓ **7.12%**	34.17 px ⇑ 9.97 px
St-Marc	0.6781 ⇓ **7.86%**	7.74 px ⇑ **1.77 px**	0.3352 ⇓ 26.66%	16.78 px ⇑ 2.20 px

number on the left of the arrow represents the results and the number on the right represents the difference between the MOTA and the MOTP for the optimized set ofparameters and the scene specific parameters. The MOTA difference is express in percent points. For the MOTA column, a U represents a lower tracking accuracy while for the MOTP column, a ⇑ represents a lower tracking precision.

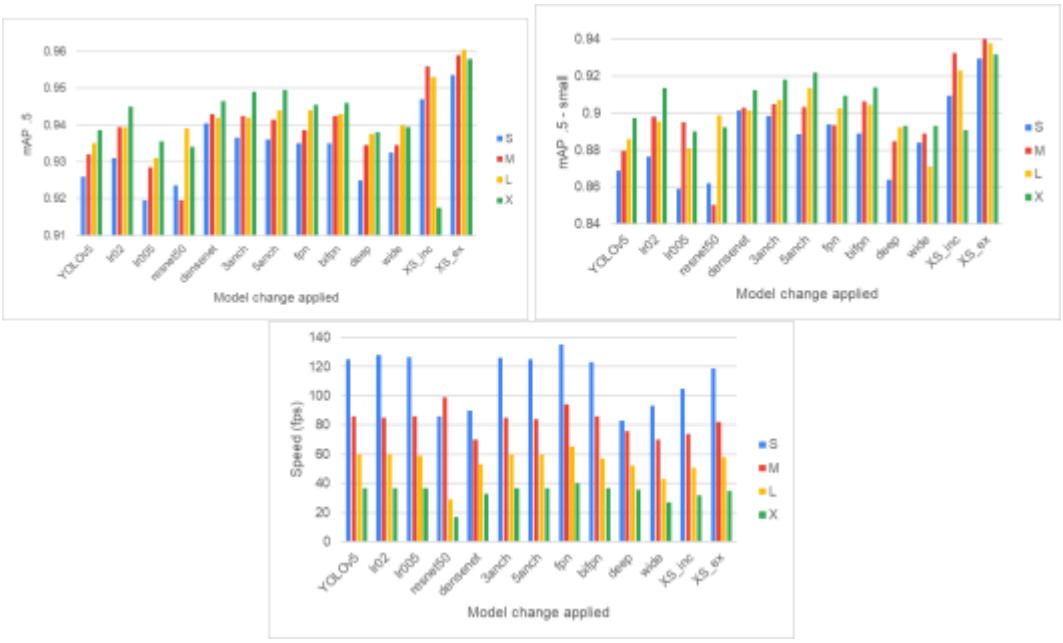
GRAPHS



This figure shows the comparison between different algorithms.



This above graph the relation between the position (in x and y value of the centrepoint). Width and height of instances of the dataset.



This above graph shows results of applying individual architectural changes toYOLOv5 at each scale**.**

CONCLUSION

In this paper, we studied a model for detecting objects in conditions that are confusing to detect objects. To create this environment, images were acquired using a drone in situations where it was confusing to detect objects such as various altitudes, weather, and background. In addition, it aimed to detect objects in these environments and increases detection performance. The experimental method is based on the YOLOv5 structure. We compared theresults with the original YOLOv5 model and improved the YOLOv5_Our model, and throughtraining, it was selected for the YOLOv5_Ours model with the best performance. Then, the best weight obtained through validation is applied to the YOLOv5_Ours model and tested.
As a result, we found that the mAP has increased to 0.9% compared with the original YOLOv5 model and improved the YOLOv5_Ours model. Finally, for a more accurate comparison, the key indicators were calculated with the previous version of YOLO: YOLOv3 and YOLOv4. The difference between the value of YOLOv3, YOLOv4, and mAP was 1.6% and4.5%, respectively, which was greater than the original YOLOv5 model. In addition, it was confirmed that the convergence speed of loss function of YOLOv5_Ours model was reduce the compared to original YOLOv5 model at the beginning of training. Object detection using drones is greatly influenced by the surrounding environment. We conducted research to improve the performance of the model under bad conditions, and we were able to obtain improved results. It may be applied to object recognition studies Appl. Sci. 2022, 12, 7255 15of 16 using drones that have been previously conducted [46,47]. In the future, the results of this study will help use drones to detect objects in various conditions. In this paper, we add some cutting-edge techniques i.e. transformer encoder block, CBAM and some experienced tricks to YOLOv5 and form a state-of-the-art detector called TPH-YOLOv5, which is especially good at object detection in drone-captured scenarios. We refresh the record of VisDrone2021 dataset, our experiments showed that TPH-YOLOv5 achieved state-of-the-artperformance in VisDrone2021 dataset. We have tried a large number of features, and used some of them to improve the accuracy of object detector. We hope this report can help developers and researchers get a better experience in the analysis and processing of drone- captured scenarios

REFRENCES

1. Z Li - Journal of Physics: Conference Series, 2022 - iopscience.iop.org
2. MP Mathew, TY Mahesh - Signal, Image and Video Processing, 2022 - Springer

3. HK Jung, GS Choi - Applied Sciences, 2022 - mdpi.com
4. F Dadboud, V Patel, V Mehta, M Bolic… - 2021 17th IEEE …, 2021 - ieeexplore.ieee.org
5. A Benjumea, I Teeti, F Cuzzolin, A Bradley - arXiv preprint arXiv …, 2021 - arxiv.org
6. Y Chen, C Zhang, T Qiao, J Xiong… - … Conference on **Graphics** …, 2021 - spiedigitallibrary.org
7. T Sharma, B Debaque, N Duclos, A Chehri, B Kinder… - Electronics, 2022 - mdpi.com
8. A Malta, M Mendes, T Farinha - Applied Sciences, 2021 - mdpi.com
9. Y Yu, J Zhao, Q Gong, C Huang, G Zheng, J Ma - Remote Sensing, 2021 - mdpi.com
10. V Sharma - 2020 - scholarworks.calstate.edu
11. Girshick R, Donahue J, Darrell T and Malik J 2014 Rich feature hierarchies for accurate object detection and semantic segmentation Proc. of the IEEE Conf.
12. J Fang, Q Liu, J Li - 2021 IEEE 6th International Conference on …, 2021 - ieeexplore.ieee.org
13. G Sun, S Wang, J Xie - Electronics, 2023 - mdpi.com
14. Z Wu, D Zhang, Y Shao, X Zhang… - … Pattern **Recognition** …, 2021 - ieeexplore.ieee.org
15. F Corputty, SA Wibowo, S Rizal - 2022 5th International …, 2022 - ieeexplore.ieee.org
16. T Huang, M Cheng, Y Yang, X Lv, J Xu - … on Image and **Graphics** …, 2022 - dl.acm.org
17. L Ting, Z Baijun, Z Yongsheng… - 2021 6th International …, 2021 - ieeexplore.ieee.org
18. Z Qu, L Gao, S Wang, H Yin, T Yi - Image and Vision Computing, 2022 - Elsevier
19. M Karthi, V Muthulakshmi, R Priscilla… - … and Smart Electrical …, 2021 - ieeexplore.ieee.org
20. M Sozzi, S Cantalamessa, A Cogato, A Kayad… - Agronomy, 2022 - mdpi.com
21. E Güney, C Bayilmiş, B Cakan - IEEE Access, 2022 - ieeexplore.ieee.org