International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Comparison of Transfer Learning Techniques for Object Detection

Ayush Majumdara, Siddharth Sarmaa, Sarvesh Satonea, Atharv Mankara, Bhavana Tiplea and Madhura Phataka

Submitted:27/03/2023 **Revised:**26/05/2023 **Accepted:**11/06/2023

Abstract: The YOLO series of models has been the industry standard for precise and practical object identification since 2015. Since then, YOLO models have been improved for faster and more accurate detection as well as meeting a variety of requirements in real-time environments and multiple pertinent scenarios. The first YOLO model introduced the concept of tackling object detection by allowing a neural network to predict bounding boxes and class probabilities in one evaluation. A novel dataset of Indian roads and vehicles has been created using a highly customised dataset of 3k photos obtained from various sources to compare and analyse YOLO models such as YOLOv5, YOLOv6, YOLOv7, and YOLOR in-depth in this paper. The study's findings reveal that, in a similar testing environment, YOLOv5 performs better than the competition, making it the most accurate YOLO Model to date.

Keywords: *Object Detection; YOLOv5; Transfer Learning; Deep Learning*

Introduction

The science of computer vision has recently experienced a growth across a wide range of businesses. The YOLO series is the most well-liked among machine learning aficionados of all the currently available object identification systems because of its quick speed and straightforward architecture, which produce results that are far more exact and accurate. The first in the YOLO series, the YOLOv1, was released in 2015 by Joseph Redmon [1]. It was revolutionary in and of itself because it proposed an improved way for object recognition. Then came YOLOv2 [2], which made a significant improvement to the architecture. Further, YOLOv4 [3] [4] was released, dividing the detection system into many components (head, neck, backbone). In the current competition for effective object detection, YOLOv5, YOLOv6 [5], YOLOX [6], YOLOR [7], PP-YOLOE [8], and YOLOv7 [9] are the contenders.

School of Computer Engineering and Technology, Dr.Vishwanath Karad MIT World Peace University, Pune

Email: ayushmajumdar6501@gmail.com (Ayush Majumdar) Email: siddharthgsarma5@gmail.com (Siddharth Sarma) Email: sarveshsatone1506@gmail.com (Sarvesh Satone) Email: atharvmankar2215@gmail.com (Atharv Mankar) Email: bhavana.tiple@mitpwu.edu.in (Bhavana Tiple) Email: madhura.phatak@mitwpu.edu.in (Madhura Phatak)

In order to make recommendations to the reader about which model would be su- perior in a particular scenario or use case, this paper compares and contrasts many versions of the YOLO series in great detail.

To ensure a common factor, the same dataset of 3k images from the web, Google V6 143 (2022b) image repository, and personally clicked camera images is used in the analysis, and the same metrics are measured across all the models mentioned above, making the individual performances of the models comparable. By comparing the performance of several YOLO models on the same dataset, the reader can learn more about each model's unique characteristics and how they differ from one another. It also assists the reader in evaluating which YOLO series line to apply in a given situation.

Background

The proposed work is to further revolutionize object detection and apply it for a unique and relevant use case in today's world. Applications such as Autonomous navigation demands robust object detection systems as a primary prerequisite and therefore it is desirable to enhance the accuracy of these models to solve unprecedented challenges. Object detection has gone through groundbreaking improvements throughout the years since its inception. However, in the context of the autonomous driving setting, new and unique challenges are still being faced that haven't yet been solved unconditionally even by the already existing leading techniques. The study has taken into account several already existing research projects in chronological order of their publication, some of which are mentioned below, on object detection algorithms and how they are implemented to accumulate essential information on how this field of study has burgeoned expeditiously, particularly in recent years, along with what the proposed model can be inspired from and what gaps it may fill based on specific objectives and needs.

Zhu, J. et al., 2012 [10] proposes an inventive multi-sensor multi-level augmented convolutional network model as an effective technique of reducing collisions, which is termed as multi- sensor multi-level enhanced convolutional network architecture (MME-YOLO).

Huansheng Song et al., 2019 [11] proposed a methodology wherein the highway road surface in the image is first extracted and separated into a remote area and a proximate area in this vision-based vehicle identification and counting system.

Tanvir Ahmad et al., 2020 [12] proposed an updated and new network based on the YoloV1 and increasing performance by incorporating optimization of the loss func- tion of YoloV1 along with effectively extracting features and adding a spatial pooling pyramid layer.

Chengpeng Wang et al., 2021 [13] proposes an improved lightweight network based on Yolov5 to meet the needs for rapid detection of vehicles on the road and state that expanding the dataset through augmentation and applying cutoff and saturation transformation increased their model accuracy by 8.5%.

Yeting Huang et al., 2021 [14] mainly compares the accuracy and efficiency of de- tection mechanisms of YoloV3 and YoloV5 and aims to establish a digital assistant driving detection system, and make excellent decision-making reactions to reduce traf- fic accidents.

The main contributions of our proposed study are to account for a much larger and more versatile dataset including multiple novel classes indigenous to the Indian Road
milieu, and optimal pre-processing to support the YOLO models to produce faster and highly accurate scores , much improved than that of its predecessors. The impact of our approach also addresses the exigent limitations, the primary of which is the lack of a well-rounded and more inclusive model for Object detection in countries like India. The study tackles the aforementioned limitation by predefining classes of object detection specific to the streets of India. Accounting for more potential entities on the streets and optimizing the model on a larger and country-specific dataset yields a more robust and conclusive model applicable to a multifaceted ecosystem. The classes in the model are not limited to only cars and pedestrians, but consider other prevalent entities such as groups of potholes, bikes, autos that are found on Indian roads, and traffic lights and signs found on the edge of lanes in urban and rural areas alike.

Methods

In the field of computer vision, object detection is a pervasive issue which deals with detecting, identifying, and localizing different classes of objects in a provided image or video. Most commonly it involves enclosing the detected and classified object with a bounding box along with stating the confidence that the deep learning model has in the prediction. This project entails a detailed use of the latest YOLO models to analyze their ability to detect different objects in real-time due to its high detection speed as well as the ability to detect objects with high frame rates which is more in conjunction with a real-world scenario. The models have been selected based on their exceptional results in terms of efficiency and precision with the COCO dataset [15] which is a large-scale object detection, segmentation, and captioning dataset.

Outline of the research:

Dataset Collection and Annotation

Dataset Pre-processing and Augmentation

Custom Dataset labeling and Annotation

Models and Metrics

Analysis of Results and Inference

Dataset Collection and Generation

The data for this study uses real-time images taken by the camera as well as those available in the public domain. A total of upto 3k images were collected. Efforts were put in by the authors to make sure that the dataset was generalized and consisted of multifarious images taken at different times of the day and lighting conditions to make way for more robust detection. Images were also selected to have a wide variety of vehicle models in different real-time scenarios. All given classes unless mentioned otherwise are a mixture of a collection of scraped images from the web using a custom python script, taken from the Google V6 [16] image repository or a collection of camera clicked pictures to bring diversity to the database:

Car
Auto Rickshaw
Pedestrian [17]
Potholes [18]
Bikes
Traffic Light

Dataset Pre-processing and Augmentation

Image pre-processing is used for developing images with suitable properties in model training and inference. Commonly used pre-processing techniques involved are resiz- ing, orienting, color corrections, etc. Image augmentation is the process of manipulat- ing photos to form multiple varied versions of the same image in order to train the model with more examples. To get a robust image dataset for model input, image pre-processing is required. Fully connected layers in convolutional neural networks, for example, demand that all images be of the same size. Image pre-processing can also accelerate model training time and speed up model deduction considerably.

A set of augmentation techniques [19] are applied to the existing data set collected, which includes- flipping images horizontally and vertically. A rotation of 10 degrees in each direction (Clockwise, Counter-Clockwise, Upside- ° Down). °Cropping° of images with -60% minimum zoom and 140% maximum zoom was applied. Shearing is applied in the range of -30 to +30 and 90 horizontal. The size of the input images in the dataset is set to 256px x 256px. Using the data augmentation techniques allowed us to increase our dataset size to a total of 2.5k images.

Custom Dataset Labeling and Annotation

For annotations, LabelImg [20] tool was used to draw the ground truth bounding boxes. Annotations were saved as text files in YOLO format. YOLO demands each image to have annotations in the form of a .txt file, with each line describing a bounding box. Each line describes a single object and is composed of five strings: class_name, x_centre, y_centre, width, height. The class_name takes an integer value from 0 to (N-1), where N denotes the total number of classes to be detected. The remaining 4 values comprise the box coordinates and are normalized to obtain a value between 0 and 1.

Models & Metrics

YOLO

YOLOv5 is the latest in the line of YOLO (You Only Look Once) family of object detection algorithms coined by Joseph Redmon in his 2016 publication [21]. The YOLO Models are very well-known for their high-speed detection whilst coming in a comparatively smaller package. It is based on the Darknet neural network which is a C and CUDA-based open source neural network framework. Redmond's work has been continued by Alexey Bochkovskiy after YOLO's 3rd iteration [22] and resulted in the fastest real-time model for object detection in YOLOv4 [23]. YOLOv5 was then re-leased by Glenn Jocher with several significant changes. YOLOv5 was the first member of the YOLO family to be written in PyTorch rather than Darknet which resulted in simpler support as well as easier deployments.

YOLOv5

YOLOv5 [24] released by Glenn Jocher with several significant changes. It was the first member of the YOLO family to be written in PyTorch which resulted in simpler maintenance as well as easier deployments. Its structure comprises a backbone, neck, and head. Cross Stage Partial Networks are the backbone and it extracts the features from the input image and is used as the neck which features pyramids and helps to identify the same object with different scaling. Lastly, the head in particular is used to carry out the very last and important detection part. YOLO algorithms split all of the input images using the SxS grid structure. Object detection is the obligation of every grid. The boundary boxes for the observed object are now predicted by those grid cells. We have five major attributes for each box: x and y for coordinates, w and h for object width and height, and a confidence score for the possibility that the box encompasses the object. [25]

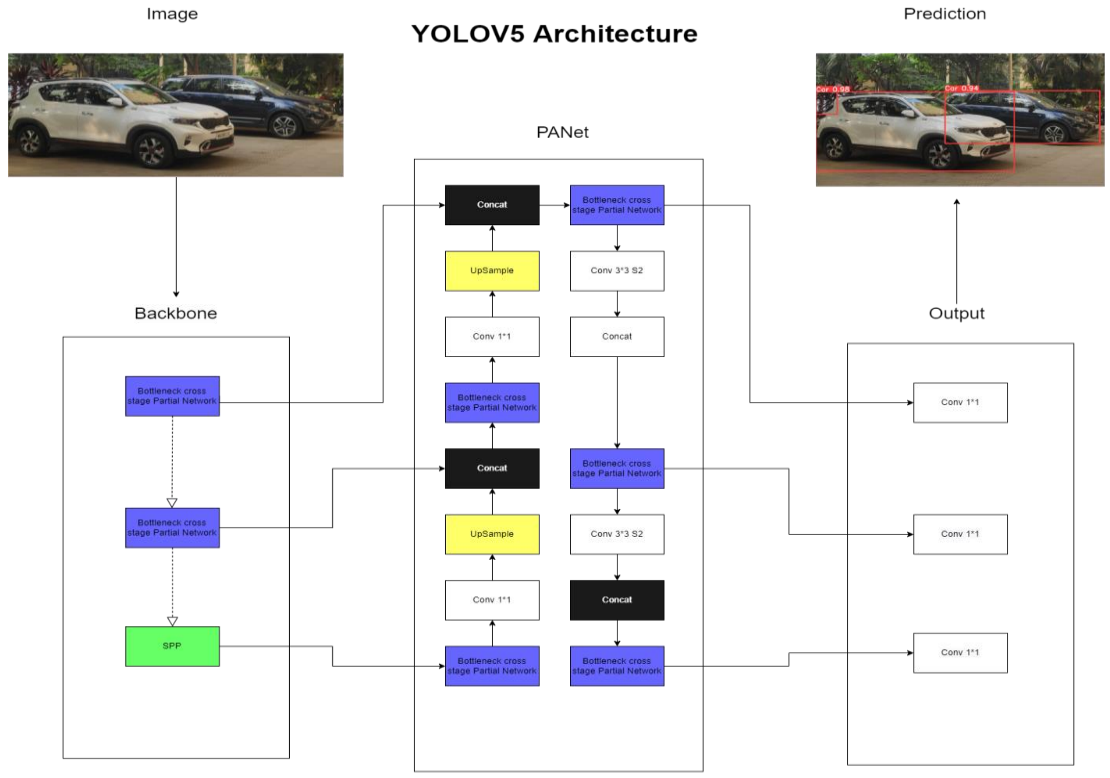

Fig 1. Structure of YOLOv5 model - Backbone, Neck and Head Alt Text: A figure showing the YOLOv5 architecture including the input and output images

MT-YOLOv6 was given the moniker YOLOv6 by its creators because it was mod- eled after the original one-stage YOLO architecture. It's vital to remember that MT- YOLOv6 is not a part of the official YOLO series although it is said to produce better

results. In comparison to all previous YOLOv5 variants, YOLOv6s offer a better mean Average Precision (mAP) and an approximately 2-times faster inference time. Yolov6 in its architecture backbone uses RepBlock [26] as the fundamental building block of the small network. The neck adopts the PAN [27] topology similar to its previous iterations. Herein, an efficient re-parameterizable backbone denoted as EfficientRep. Also, a hybrid-channel approach to create a decoupled head that is more effective was implemented. In particular, the number of the middle 3x3 convolutional layer is lim- ited to a single layer. Adding to these the anchor point-based paradigm, whose box regression branch actually forecasts the separation between the anchor point and the four bounding box sides were used.

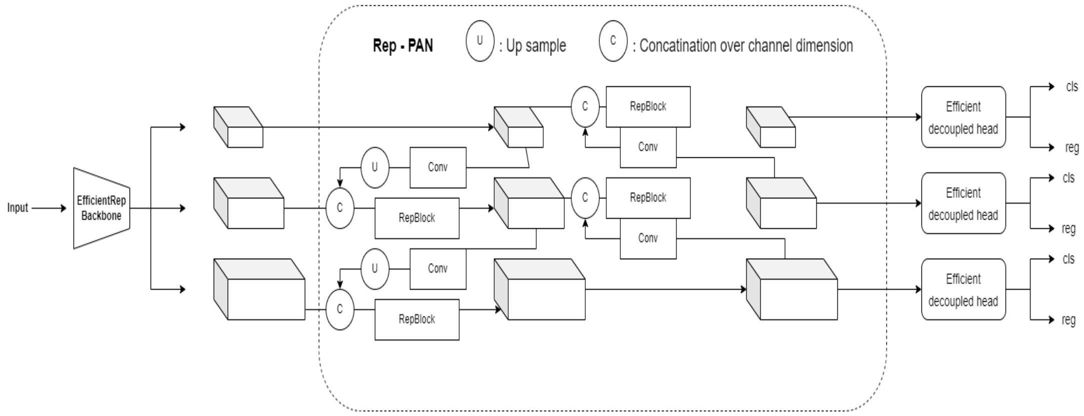

Fig 2. The YOLOv6 architecture Alt Text: A figure deep diving into EfficientRep Backbone of the YOLOv6 architecture

YOLOv7

YOLOv7 outperforms all other known object detectors in the range of 5 FPS to 160 FPS in terms of both speed and accuracy, and it has the greatest accuracy of 56.8% AP among all real-time object detectors with 30 FPS or above. In general, YOLOv7 offers a quicker and more robust network architecture that offers a better feature in- tegration approach, more precise object recognition performance, a more robust loss function, and an improved label assignment and model training efficiency. Because of this, YOLOv7 uses far less expensive computational hardware than other deep learning models. Without any pre-learned weights, it can be trained significantly more quickly on tiny datasets. The authors of YOLOv7 build on prior research on the subject while taking into account the memory requirements for maintaining layers in memory and the distance over which a gradient can propagate back through the layers; the shorter the gradient, the more effectively their network will be able to learn. They settle on E-ELAN, an extended version of the ELAN computational block, as their final layer aggregate.

YOLOR

A novel object detection technique called YOLOR was introduced in 2021, and it matches and even exceeds a scaled YOLO v4 model. As it employs a single network to simultaneously capture implicit and explicit knowledge, YOLOR is conceptually distinct from YOLO. In a convolutional neural network, YOLOR can do "kernel space alignment, prediction refinement, and multi-task learning," and the authors' research shows that using implicit knowledge improves how well all tasks are performed. YOLOR is especially designed for object detection as opposed to other machine learning use cases like object analysis or identification. The authors propose a unified network that is capable of performing a variety of tasks. This network integrates implicit and explicit knowledge to create a general representation that is then used to do a variety of jobs. The model's performance is effectively enhanced by the suggested network at very little additional cost.

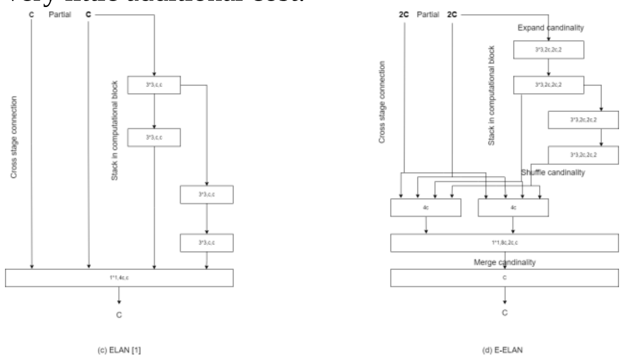

Fig 3. YOLOv7 ELAN and E-ELAN architecture Alt Text: Figure comparing the ELAN and the E-ELAN architecture, the latter of which is used a final layer aggregate in YOLOv7.

Transfer Learning

Transfer Learning can be defined as a Machine Learning technique that involves train- ing and developing a model for one task and subsequently reusing it on a related task [28]. It describes a scenario in which acquired knowledge in one context is applied to gain optimized results in another.

Transfer Learning is used to train a pre-trained model using a customized, fresh dataset that is smaller in size compared to the origi- nal dataset. Transfer learning is a method of boosting a learner's performance in one domain by transferring knowledge from another similar or related domain [29]. We can discover why transfer learning is possible by looking at real-world non-technical experiences. Consider an amateur and a deft badminton player. The skilled player will be able to learn another similar game such as table tennis much more efficiently than the amateur player by transferring his already acquired information to the new task of learning table tennis.

This paper proposes a system which uses the YOLO [24] pre-trained weights and is currently being re-purposed to learn new features(or transfer them), to be further trained on our custom dataset and classes. This allows us to begin with the learned weights and adjust the newer or different features accordingly instead of initiating the learning process on the dataset from scratch which reduces training time considerably as well as boosts accuracy. This can be seen in the results of our machine learning model where a robust object detection system with a precision of 93.5% was trained in just 5.637 hours.

Metrics

Mean Average Precision(mAP) is an object detection metric that is used to evaluate the precision of a model's object detection precision compared to ground-truth object annotations in the provided dataset. It is often used as the default metric to calculate the accuracy obtained of object detection models like the Yolov5. Mean Average Preci- sion can be better understood by breaking down the terms involved. Precision reflects the measure of accuracy in the true predictions made by the model for a particular class. This can be described by the following (for a particular class):

Precision = [30]

True Positive

True Positive+False Positive

Another parameter that work in hand with precision is Recall which is described as:

Tru*Recall* = [30]

e Positive True Positive + False Negative

However to define what classifies a correct prediction we need another metric known

as Intersection over Union (IoU). *Area of Union*

Intersection over Union = Area of Intersection [30]

It provides us with the overlap measure between the predicted and the ground truth bounding boxes. The more the overlap/intersection, the better the prediction. Gener- ally, an IoU threshold is set, say 0.5, to determine whether the classified prediction is a true positive. We can now define Average Precision as the weighted mean of precision acquired at each confidence level of the model, with the increase in recall from the prior threshold as the weighting factor. Thus, mAP can then be defined as the mean of all such Average Precisions at different IoU thresholds.

Loss Function

The loss function estimates the difference in the algorithm's current and expected outcome [30]. The loss calculated is a compound loss based on the Objectness Score, Bounding Box Regression Score and Class Probability. Binary Cross-Entropy Loss is used for class probability whereas Generalized IoU (GIoU) Loss acts as the loss for bounding box regression score. It is calculated as follows:

ttIoU = Area of Intersection − |C/Area of Union|

Area of Union

|C|

Results

All of the aforementioned algorithms were trained on the dataset for 150 epochs each with a batch size of 16, 32 and 64 to determine the correlation between the batch sizes and the accuracy. The YOLOv5 model achieved the highest mAP score of 93.7%. Throughout the course of the experiments, comparative intermediate weights have chosen so as to keep the comparison as fair as possible.

Table 1. Training results on 150 epochs

Algorithms	Batch 16	Batch 32	Batch 64
YOLOR	93.00	92.00	90.4
YOLOv5	93.5	93.10	93.70
YOLOv6	89.47	88.50	88.23
YOLOv7	93.6	90.1	89.4

The YOLOv5 model returned the highest accuracy of 93.7% with a batch size of 64, which was closely followed by YOLOv7 with the second highest accuracy of 93.7% but with a batch size of 16. For all the four algorithms, the model's mAP for the output at a threshold of 0.5 IoU seemed to reach the saturation point towards the end of the training with fluctuations throughout. YOLOR follows closely behind YOLOv7 with mAP score of 93% with a batch size of 16, and YOLOv6 concludes as the least accurate of the models with the highest score of 89.47 with a batch size of 16 which is 0.53 behind YOLOR and 4.23 behind the leader YOLOv5.

Results across the board have concluded that smaller batches have contributed to a higher mAP score except for the anomalous YOLOv5 where batch 64 ended up being the most accurate, albeit with a difference of 0.2.

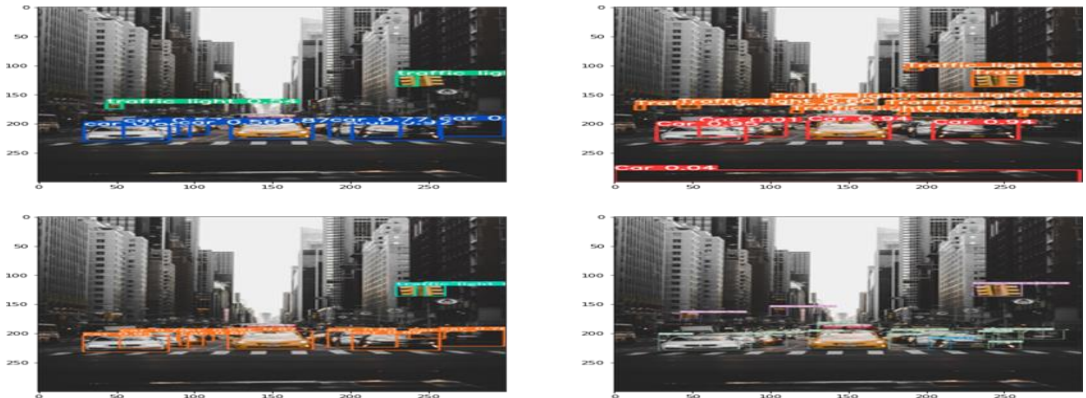

Fig 4. From top left to bottom right, i) YOLOR ii) YOLOv5 iiI) YOLOv6 iv) YOLOv7 Alt Text: 4 different algorithms detecting objects of similar classes in the same scenario, helping us to validate the results and select the best

Figure 4 showcases the inferences of all the four algorithms in a real world scenario. We can clearly observe that YOLOv5 has detected multiple traffic lights which are very minute and almost invisible to the naked eye whereas the other three algorithms have only detected the predominant and clearly visible traffic light. Detection of cars is fairly even throughout the images and can be considered as a stalemate. Confidence of detection too is the highest in the case of YOLOv5.

Fig5. From top left to bottom right, i) YOLOR ii) YOLOv5 iii) YOLOv6 iv) YOLOv7 Alt Text: 4 different algorithms detecting objects of different classes in a different lighting scenario, helping us to validate the results and select the best algorithm.

Figure 5 not only extends the previous conclusion but also cements YOLOv5 as the clear winner. All four algorithms have detected the cars as well as the pedestrians with relative ease but YOLOv5 manages to do it with the highest confidence.

Conclusion

This study compared the most recent and sophisticated object detection machine learn- ing models from the YOLO series. With thousands of photos being analysed to make choices in real-time across a range of industries, including driver-less vehicles, indus- trial manufacturing, and imposing greater security, to name a few, the importance of object detection is plain for anyone to notice in today's world. The same customised and varied dataset was used to train all the algorithms, and it was discovered that when the models were trained on a batch size of 16, YOLOv7 offered the highest accuracy with 93.8%. YOLOv5 offered the highest accuracy with 93.1% and 93.7%, respectively, for a batch size of 32 and 64. Therefore, it can be said that YOLOv5 exhibits the best overall performance out of the four YOLO models examined.

There is still a ton of work to be done in this area in the future. New algorithms or revisions to current ones are published every year. Additionally, different algorithms are best suited for different industries, including agriculture, terrestrial and aerial au- tonomous vehicles, industrial machines, aviation etc. In-depth examination of these topics is possible in the future.

Competing Interests

All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Data Availability

The data that support the findings of this study are available from the corresponding authors, upon reasonable request.

References

[1] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection . 2016.

[2] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement . 2018.

[3] Wang CY, Bochkovskiy A, Liao HYM. Scaled-YOLOv4: Scaling Cross Stage Partial Net- work . 2021.

[4] Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal Speed and Accuracy of Object Detection . 2020.

[5] Li C, Li L, Jiang H, et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications . 2022.

[6] Ge Z, Liu S, Wang F, et al. YOLOX: Exceeding YOLO Series in 2021 . 2021.

[7] Wang CY, Yeh IH, Liao HYM. You Only Learn One Representation: Unified Network for Multiple Tasks . 2021.

[8] Xu S, Wang X, Lv W, et al. PP-YOLOE: An evolved version of YOLO . 2022.

[9] Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: Trainable bag-of-freebies sets new state- of-the-art for real-time object detectors . 2022.

[10] Zhu J, Li X, Xu JP, et al. Multi-Sensor Multi-Level Enhanced YOLO for Robust Vehicle Detection in Traffic Surveillance. Sensors. 2012;.

[11] Zhu J, Li X, Jin P, et al. MME-YOLO: Multi-Sensor Multi-Level Enhanced YOLO for Robust Vehicle Detection in Traffic Surveillance. Sensors. 2020 12;21:27. Available from: 10.3390/s21010027.

[12] Ahmad T, ma Y, Yahya M, et al. Object Detection through Modified YOLO Neural Network. Scientific Programming. 2020 06;Available from: 10.1155/2020/8403262.

[13] Wang CY, Bochkovskiy A, yuan Liao H. Scaled-YOLOv4: Scaling Cross Stage Partial Network. 06; 2021. p. 13024–13033. Available from: 10.1109/CVPR46437.2021.01283.

[14] Y H, H Z. A Safety Vehicle Detection Mechanism Based on YOLOv5. IEEE 6th Interna- tional Conference on Smart Cloud. 2018;.

[15] Lin TY, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context. In: Fleet D, Pajdla T, Schiele B, et al., editors. Computer Vision - ECCV 2014; Cham. Springer International Publishing; 2014. p. 740–755.

[16] Open Image dataset V6+ . 2022. Available from: https://storage.googleapis.com/ openimages/web/index.html.

[17] Penn-Fudan Database for Pedestrian Detection and Segmentation. . 2007. Available from: https://www.cis.upenn.edu/\textasciitildejshi/ped_html/.

[18] Kumar A . 2020. Available from: https://www.kaggle.com/atulyakumar98/pothole- detection-dataset.

[19] Keras Image Data Preprocessing . 2022. Available from: https://keras.io/api/ preprocessing/image/.

[20] Tzutalin. LabelImg . 2007. Available from: https://github.com/tzutalin/labelImg.

[21] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection . 2016.

[22] Vinston Raja, R., Ashok Kumar, K. ., & Gokula Krishnan, V. . (2023). Condition based Ensemble Deep Learning and Machine Learning Classification Technique for Integrated Potential Fishing Zone Future Forecasting. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2), 75–85. https://doi.org/10.17762/ijritcc.v11i2.6131

[23] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. arXiv. 2018;Available from: https://arxiv.org/abs/1804.02767.

[24] Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv. 2020;Available from: https://arxiv.org/abs/2004.10934.

[25] Ultralytics. Ultralytics Yolov5 . 2022. Available from: https://github.com/ultralytics/ yolov5/releases.

[26] Xu R, Lin H, Lu K, et al. A Forest Fire Detection System Based on Ensemble Learning. 2021;12:217. Available from: 10.3390/f12020217.
[27] Chang Lee, Deep Learning for Speech Recognition in Intelligent Assistants , Machine Learning Applications Conference Proceedings, Vol 1 2021.
[28] Ding X, Zhang X, Ma N, et al. Repvgg: Making vgg-style convnets great again. Proceed- ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021; :13733–13742.
[29] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018;:8759–8768.
[30] Hussain M, Bird JJ, Faria DR. A Study on CNN Transfer Learning for Image Classifi- cation. In: Lotfi A, Bouchachia H, Gegov A, et al., editors. Advances in Computational Intelligence Systems; Cham. Springer International Publishing; 2019. p. 191–202.
[31] Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. Journal of Big Data. 2016 May;3(1):9. Available from: 10.1186/s40537-016-0043-6.
[32] Wang Q, Ma Y, Zhao K, et al. A Comprehensive Survey of Loss Functions in Machine Learning. Annals of Data Science. 2022 04;9. Available from: 10.1007/s40745-020-00253-5.