2024 International Conference on Electrical, Computer and Energy Technologies (ICECET) | 979-8-3503-9591-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICECET61485.2024.10698500

Deep Learning Application for Vehicle Detection through Surveillance Drones

Abeer Ilyas
Department of Computer Science
Institute of Space Technology
Islamabad, Pakistan
abeer.ilyas2001@gmail.com

Imama Rahmani
Department of Computer Science
Institute of Space Technology
Islamabad, Pakistan
imamarahmani@outlook.com

Sukaina Imran
Department of Computer Science Institute of Space Technology
Islamabad, Pakistan
sukainaimran25@gmail.com

Tufail Sajjad Shah Hashmi
Department of Computer Science
Institute of Space Technology
Islamabad, Pakistan
stufail110@gmail.com

Muhammad Nadeem Yousaf
Department of Computer Science
Institute of Space Technology
Islamabad, Pakistan
nadejoseph11@gmail.com

***Abstract*—Vehicle detection plays an essential role in a variety of real-world applications, including autonomous driving, traffic surveillance, and management. This research work thoroughly compares the three deep learning models for vehicle detection—You Only Look Once (YOLOv8), Faster Region-Convolutional Neural Network (R-CNN) and Single Shot MultiBox Detector (SSD). The study is conducted using different datasets, especially the Pak Vehicles Dataset, which consists of diverse images taken in different environmental situations across Pakistan. It also includes a description of annotation methods and explains the architecture of each model. The evaluation focuses on accuracy, speed, and robustness metrics like Mean Average Precision (mAP), recall, and inference time. The result highlights strengths, weaknesses, and insights for improving real-world vehicle detection systems. For full implementation of these models, kindly check this: https://shorturl.at/nJRSV**

***Index Terms*—Vehicle detection, Pakistan Vehicle Dataset, UAV, YOLO, Fastest R-CNN, SSD, Comparative Analysis.**

I. Introduction

The evolution in deep learning models has significantly improved the proficiency and performance of object detection, a crucial aspect of computer vision with vast applications. One of the fundamental applications is vehicle detection, which is important in various domains, including traffic accident investigation, autonomous driving, traffic surveillance [1], intelligent transportation systems (ITSs) [2, 3], public safety and security systems. In the diverse scope of computer vision, vehicle detection is a vital application. Detecting and categorising vehicles as objects from videos and images is complicated using appearance-based representation, yet it plays a key role in the real-time applications of vehicle detection. Many object detection models have been developed that can be used for the purpose of vehicle detection. Therefore, selecting a suitable model plays an essential role in achieving efficient and accurate results. This study provides a comprehensive comparative analysis of three state-of-the-art (SOTA) object detection models: You Only Look Once (YOLOv8), Faster Region-based Convolutional Neural Network (R-CNN), and Single Shot Detector (SSD). These models use a variety of approaches, frameworks, and procedures to produce precise and effective outcomes. This analysis aims to provide invaluable insights into the strengths, limitations, and performance metrics of the latest object detection models by assessing their precision and accuracy.

A. Motivation

The motivation behind this research is to study, analyse and select the most appropriate object detection model for vehicle detection through Unmanned Aerial Vehicles (UAV). The scope and vision are to enhance traffic surveillance and monitoring, parking management, vehicle speed control, public safety and security.

B. Main Contributions

This research contributes to the field of computer vision by introducing the application of vehicle detection using the unique dataset. No prior research has discussed vehicle detection explicitly customised to the Pakistani environment, covering many vehicles, including rickshaws, cars, buses, bikes and trucks. By using this Pak Vehicles Dataset, we reveal new perspectives on the performance and challenges of vehicle detection within this particular regional context. This study fills a large void in the literature and offers a substantial contribution to practical applications such as urban planning, vehicle surveillance and traffic control in Pakistan. This study sets up a foundation for future computer vision research projects specific to Pakistan's automobile landscape. This will

979-8-3503-9591-4/24/$31.00 ©2024 IEEE

also help the developers and researchers make knowledgeable decisions when selecting an object detection algorithm for vehicle-related applications.

C. Paper Structure

The remaining sections of this paper are as follows: Section II outlines the related work. Section III explains the experimental methodology, along with dataset selection, evaluation metrics, architecture and parameters of the models, and hardware/software configurations. Section IV shows the implementation of the models. Section V examines the performance of the trained models and displays the results. Section VI concludes the paper.

II. Literature Review

Several research studies have previously compared the efficacy and performance of deep learning models in this domain, highlighting both their benefits and drawbacks.

One relevant research study comparing YOLOv8, YOLOv5, Faster R-CNN, and EfficientDet for remote sensing was carried out by Rustem Glue in May 2023 [4]. Findings revealed that with a mAP50 of 0.62, YOLOv8 was the fastest and most accurate of the four algorithms examined. YOLOv5 was slightly less accurate, EfficientDet was slower, and Faster R-CNN had the lowest performance. Additionally, in Satya Prakash Yadav's 2023 study, Faster R-CNN outperformed other models in terms of recall, accuracy, precision, and loss for object detection using the Roboflow Public Chess Piece datasets. However, YOLO was found to be best suited for distinguishing similar objects, continuous motion, and low image quality [5]. Furthermore, in May 2021, researchers conducted a comparative analysis of the three popular deep learning image detection algorithms using Microsoft's COCO dataset. While Faster R-CNN had good detection accuracy, its detection rate was slower. YOLOv3, with a higher mAP of 76.1 and a faster detection rate, was chosen for vehicle detection due to its effectiveness at identifying small objects [6]. Adel Ammar's December 2021 paper compared object detection techniques on two UAV imaging datasets. Faster R-CNN had the best balance between AP and inference speed, outperforming YOLOv4 by 52% in AP and being only 10% slower. However, YOLOv4 showed the best trade-off on the PSU dataset, outperforming Faster R-CNN by a factor of 2.4 and increasing accuracy by 31% [7]. Moreover, Jeong-ah Kim carried out research for Real-Time Vehicle Type Recognition in November 2020 [8]. He compared three algorithms - YOLO, Faster R-CNN, and SSD. Each algorithm was trained using a car training dataset to determine the most effective model and the type of vehicle. The findings showed that, with 93% accuracy, the YOLOv4 model outperformed the other two.

The comparative assessment of the performance of various deep learning algorithms on object detection was reviewed. For example, the identification of cars from aerial images [9], vehicle type detection [10, 11], agricultural greenhouse detection [12], real-time identification of pills [13], object/image detection [14, 15], detection of drones [16], the leguminous seed detector for smart farming [17], car detection using UAVs [18] and the real-time identification of vehicles regarding the ITSs. YOLO models, specifically YOLOv3 and YOLOv4, show the best results in both accuracy and efficiency among other models in vehicle detection, agricultural greenhouse detection, pill identification in real-time, and leguminous seed detection. If using ITSs and detecting vehicles, the YOLOv7 provides the best results, as it demonstrates high accuracy and nearly real-time operation. Although Faster R-CNN tends to be more accurate, it is slower when compared to the YOLO models, making it the choice where precision is prioritised. In scenarios where a balance is required, SSDs are implemented.

III. Materials and Methods

For this research, the following datasets, hardware, software tools and models were used to find out the best model for vehicle detection.

A. Dataset

For the experiment, three datasets are used: The Vehicle Detection Dataset [19], the VISDRONE Dataset [20], and the Pak Vehicles Dataset [21]. The Vehicle Detection Dataset and VISDRONE Dataset provide consistent images of vehicles in various instances and orientations. On the other hand, the Pak Vehicles Dataset was self-prepared by fragmenting the drone videos of the traffic situation in Sialkot, Pakistan, into frames and labelling them using Roboflow. This dataset has only one class: 'Vehicles'. This dataset provides an opportunity to examine the performance of the three models in detecting vehicles in traffic congestion situations. Other modes of transportation, such as bikes and rickshaws, were also labelled but not explicitly included in the other two datasets. Table I shows that the Vehicle Detection Dataset contains 2,740 images with 12,179 instances of the 'Vehicles' class. Meanwhile, the VISDRONE Dataset has 8,127 images with 194,361 instances of the 'Vehicles' class. The Pak Vehicles Dataset has 369 images and 11,819 instances of the 'Vehicles' class. The size (640x640) of the images is uniform throughout both datasets. The following Table I displays the splitting of training, testing and validation image and instance counts in Vehicle Detection, VISDRONE, and Pak Vehicles datasets:

TABLE I. Split of Image and Instance Counts

Dataset Names		Train	Test	Valid
Vehicle Detection	Images	1,914	276	550
	Instances	8,546	1,321	2,312
VISDRONE Dataset	Images	6,366	605	1,216
	Instances	151,682	14,596	28,083
Pak Vehicles Dataset	Images	258	36	75
	Instances	8,182	1,198	2,439

For training YOLOv8, data is labelled in YOLO format, which lists content such as paths to training/validation/testing directories, class details, and other content, and is stored in a ".yaml" file. The labelling detail of each image is stored in ".txt" file. The bounding box template for the YOLO format is:

class, *x*-centre, *y*-centre, width, and height. “Class” is a non-decimal numerical value representing the index of category N, with an index from 0 to *N*—1. “*x*-centre” and “*y*-centre” give the information of the positioning of the detected objects in the image, while “width” and “height” represent the height and width of the detected object. Conversely, the dataset is labelled in COCO format to train the Faster R-CNN and SSD. A data directory in COCO format is represented by a JSON file containing image details, descriptions, and other information about the file. JSON files have three main elements: images, annotations, and categories. The images element contains the list of IDs, names, and sizes of all images in the dataset. The annotation contains the bounding box coordinates of all desired objects in the image. The bounding box syntax of the COCO format is [*x*, *y*, width, height], where ‘*x*’ and ‘*y*’ inform about the top-left corner of the bounding box. Train, validation, and test lists each have their JSON files. Data were obtained from Roboflow Universe [22], an open-source platform containing thousands of image datasets in various formats.

B. Hardware and Software tools

The system that we used to perform this experiment had the following hardware configuration:

- Processor: Intel Core i5-13420H
- RAM: 24 GB
- Graphics Card: NVIDIA GeForce RTX 3050 (6 GB)
- OS: Windows 11

Visual Studio Code was used as an Integrated Development Environment (IDE).

C. Models

1) YOLO:

YOLO is renowned for recognising objects in real-time. Class probabilities and bounding boxes are predicted by this single-stage object detection approach in a single pass.

- YOLOv8 Architecture:

It obtains a fixed-size input image and consists of various convolutional layers as demonstrated in Fig.1. For the backbone, it uses a deep convolutional neural network (CNN), for example, Darknet, ResNet, or CSPDarknet (default). High-dimensional features are produced from the input image using the backbone, which captures several levels of abstraction. It provides a tensor as output, representing the detected objects, their class probabilities and bounding boxes [23], [24] [25], [26].

- Mathematical Function:

YOLOv8 uses advanced mathematical functions, including the Mish activation function, which is automatically integrated and enhances the model’s performance by adjusting the activation of neurons based on their input: $F(x) = x\ tanh(\text{softplus}(x))$.

- Parameters:

1) Input Image Size: Determines the quality of images processed, like how sharp or detailed they appear.
2) Batch Size: Controls how many images the model handles simultaneously during training, affecting its speed and memory usage.
3) Learning Rate: Determines how much a model adjusts its weights during training, affecting both the learning process’s speed and stability.
4) Anchors: These are predefined sizes for detecting objects, ensuring accurate predictions at various scales.
5) Confidence Threshold: Filters out detections with low confidence scores, improving the model’s precision.
6) Non-Maximum Suppression (NMS): Helps remove overlapping bounding boxes to avoid redundant detections, among other parameters.

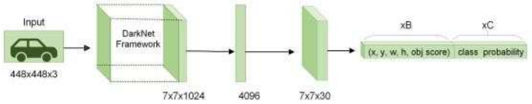

Fig. 1. YOLO Architecture

2) SSD:

Another one-stage object detection model made for real-time processing is SSD. It achieved better accuracy compared to the initial versions of YOLO [27].

- Architecture

SSD utilises a multi-scale feature extraction process and multiple convolutional layers to predict class probabilities and bounding boxes as shown in Fig. 2. After making predictions, NMS is applied to eliminate duplicate bounding boxes, retaining only the most confident detections. It uses anchor boxes of varying scales and aspect ratios to forecast object positions and shapes across diverse feature maps.

- Mathematical Function

The sigmoid function transforms raw scores into probabilities for object class predictions. Formula: $\sigma(x) = \frac{1}{1+e^{-x}}$

- Parameters

1) Input Image Size: Specifies the resolution of input images processed by the model.
2) Number of Classes: Defines the object categories that the model can identify.
3) Anchor Boxes Parameters: Include scales, aspect ratios, and positions on feature maps.
4) Learning Rate Scheduler: Utilises strategies like ReduceLROnPlateau for adjusting learning rates dynamically.
5) Optimiser: Commonly uses SGD or similar optimisers during model training.

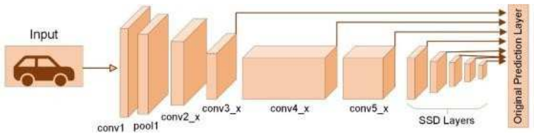

Fig. 2. SSD Architecture

3) *Faster R-CNN:*

While Faster R-CNN is a two-stage object detection algorithm with high accuracy, it requires more computation than YOLO and SSD. It uses a Region Proposal Network (RPN) to create region proposals and a subsequent network to predict bounding boxes and class labels [28], [29].

- Architecture

Faster R-CNN uses a more complex architecture with a backbone network (usually based on MobileNet) for feature extraction, a Fast R-CNN network, and an RPN for final detection as shown in Fig. 3. MobileNetV3-Large backbone includes special blocks that help it better understand different parts of the picture. It combines and understands features from different parts of the image. Region of Interest (RoI) Heads manage details for specific regions identified by the RPN and refine predictions about objects found in those regions.

- Mathematical Function

The sigmoid function helps in identifying the objects in the image and converts raw guesses into probabilities.

Formula: $\sigma(x) = \frac{1}{1+e^{-x}}$

- Parameters

1) Input Image Size: Sets the picture quality.
2) Number of Classes: Tells the model how many different objects it should recognise.
3) Anchor Boxes: Guides the model on how to search for objects.
4) Learning Rate: Controls how quickly the model learns from its mistakes.
5) Backbone Weights: Gives the backbone network its initial knowledge.
6) Trainable Backbone Layers: Decides how much the backbone can learn from new data.

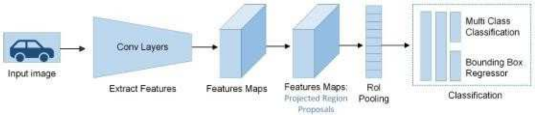

Fig. 3. Faster R-CNN Architecture

IV. Implementation Of The Models

The purpose of this experiment is to see which model is lightweight yet effective for vehicle detection from an aerial view perspective. For this reason, small pre-trained backbones were chosen, like CSPDarknet53, MobileNet V3 Large, and MobileNet V3 Large 320 FPN. CSPDarknet53 is an enhanced version of Darknet53 that utilises the Cross Stage Partial (CSP) Network to boost the model's effectiveness and performance. It allows better performance even with fewer computational resources, making it a better choice for enabling object detection and classification on edge devices. Meanwhile, MobileNet V3's Large backbone focuses on optimisation, speed, and accuracy using low computational power. MobileNet V3 Large 32 FPN uses a Feature Pyramid Network (FPN) for feature extraction from images at various scale levels with accuracy. For implementation, YOLOv8s' pre-trained weights were used to train YOLOv8, which utilises CSPDarknet backbone, MobileNet V3 Large 320 pre-trained weights for SSD training, and MobileNet V3 Large 320 FPN pre-trained weights for Faster R-CNN training [30]. Apart from the backbone configuration, every model was trained for 300 epochs per dataset using a batch size of 16 and a learning rate 0.01. The learning rate scheduler is set to ReducLRonPlateau, momentum is set to 0.9, weight decay is set to 0.0005, and the optimiser SGD is set for Faster R-CNN and SSD. The tolerance is set to 3, the factor is set to 0.1, and the mode is set to maximum for learning rate scheduling. Determining the adjusted mAP, which rises with training, is the goal of the learning rate scheduler. The YOLOv8 optimiser and learning rate scheduler are set to their default values. All three models have predefined additional training models.

V. Performance Evaluation

Recall, mAP@0 50, mAP@50-95, and inference time were chosen to analyse the performance of the trained models. The term "Mean Average Precision at 50% IoU", or "mAP@0-50", describes how well a model detects the object when it overlaps and only 50% of the target item is visible. mAP@50-95 is a comprehensive metric that evaluates a model's ability to detect a target object when it is hardly visible because of overlapping. Moreover, recall refers to the model's capacity to identify the target items precisely and reliably within the frame. Finally, inference time refers to the time taken by a model to compute an input and produce results.

A. *Result*

Table II shows the results of the trained models. YOLOv8 has the highest mAP@0-50, mAP@50 95, and recall for the Vehicle Detection Dataset, followed by Faster R-CNN and SSD. The same hierarchy held true for the VISDRONE Dataset and the Pak Vehicles Dataset. Additionally, all models performed best on the Vehicle Dataset. Overall, YOLOv8 performed better than both Faster R-CNN and SSD in the evaluation. The below Fig. 4, Fig. 5, and Fig. 6 show the inference results after assessing the attained models on an image from the test subset of Pak Vehicles Dataset.

TABLE II. PERFORMANCE METRICS OF THE MODELS

Models & Dataset		**mAP(0.5)**	**mAP(0.5:0.95)**	**Recall**
YOLOv8	Vehicle Detection	0.946	0.876	0.896
	VISDRONE	0.743	0.466	0.669
	Pak Vehicles	0.766	0.336	0.699
Faster R-CNN	Vehicle Detection	0.895	0.754	0.8
	VISDRONE	0.131	0.053	0.09
	Pak Vehicles	0.23	0.073	0.124
SSD	Vehicle Detection	0.824	0.531	0.674
	VISDRONE	0.067	0.025	0.120
	Pak Vehicles	0.152	0.042	0.135

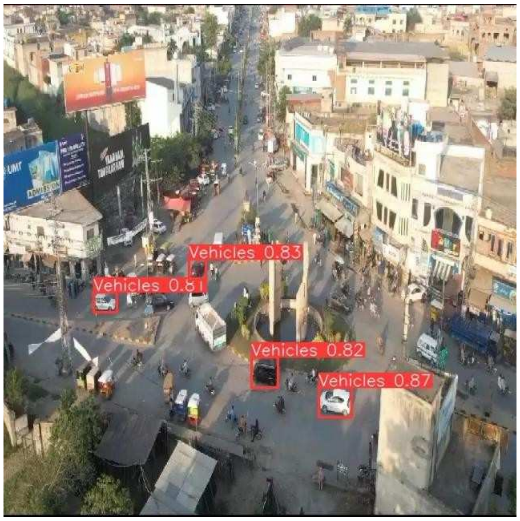

Fig. 4. YOLOv8

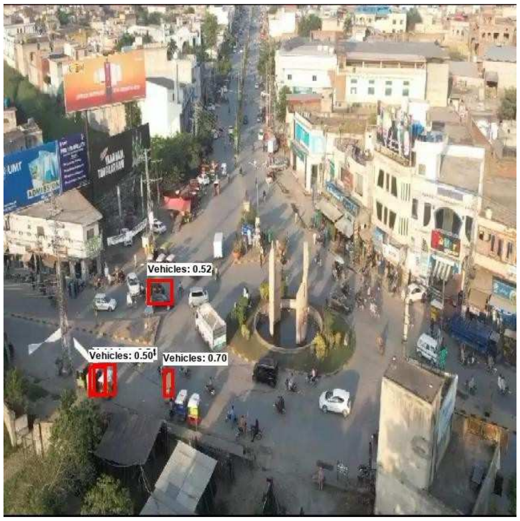

Fig. 5. SSD

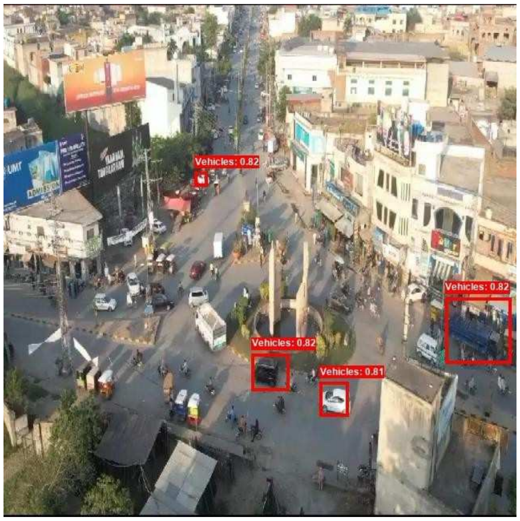

Fig. 6. Faster R-CNN

A confidence threshold of 80% was applied for YOLOv8 and Faster R-CNN inference and 50% for SSD inference as it could not detect any object with a higher confidence threshold. YOLOv8 and Faster R-CNN detected all the in-frame, visible vehicles with a high confidence score, unlike SSD. In this evaluation, the inference time of YOLOv8, Faster R-CNN, and SSD was noted to be 14ms, 116ms, and 242ms, respectively. Thus, YOLOv8 detected vehicles quickly with slightly higher accuracy as compared to Faster R-CNN and showed dominance against SSD.

B. Discussion

It is noticeable from the experiment that YOLOv8 is a better choice for vehicle detection from an aerial view than Faster R-CNN and SSD. The YOLOv8 proved to be a better model with high accuracy and less inference time, making it an ideal option for real-time applications related to vehicle detection through UAVs. Upon comparing this comparative analysis with the findings in the literature review, it can be stated that this study aligns with it, proving the superior capabilities of YOLO models. The performance of YOLOv8, being better, supports the literature review's emphasis on YOLO models for tasks like vehicle detection, agricultural greenhouse detection, pill identification, identification of different objects in GIS, identification of chess pieces and ITSs.

VI. Conclusion

In this study, the most advanced SOTA object detection models were evaluated using aerial and UAV cameras to detect vehicles. First, the datasets were attained, analysed, and pre-processed. Then, the backbones and weights were chosen for the training process, followed by setting the fundamental parameters. After repeating the training process on the first two datasets, these models were evaluated using an image attained from a test set of the Pak Vehicles Dataset. This research shows how well the YOLOv8 object detection model works, allowing it to be widely used in various industries for computer vision tasks. This research highlights the convolutions of object detection models in places where vehicles are congested or small. This also gives new researchers a stepping stone and a fundamental guide to begin with in this field. Future work includes fine-tuning these models and deploying them on edge devices and UAVs.

References

[1] H. S. Bedi, P. K. Malik, R. Singh, R. Singh, and N. Bisht. Traffic surveillance using computer vision and deep learning methods. In *IOP Conference Series: Earth and Environmental Science*, volume 1285, page 012018. IOP Publishing, 2024.

[2] D. Shokri, C. Larouche, and S. Homayouni. A comparative analysis of multi-label deep learning classifiers for real-time vehicle detection to support intelligent transportation systems. https://www.mdpi.com/2624-6511/6/5/134, 2023.

[3] E. Shurdhaj and U. Christian. Real time vehicle detection for intelligent transportation systems. https://www.diva-portal.org/smash/record.jsf?pid=diva2

[4] Rustem Gal. YOLOv8, EfficientDet, Faster R-CNN, or YOLOv5 for Remote Sensing. https://medium.com/@rustemgal/yolov8-efficientdet-faster-r-cnn-or-yolov5-for-remote-sensing-12487c40ef68 , May 2 2023.

[5] S. P. Yadav, M. J., P. Rani, V. H. C. de Albuquerque, C. dos Santos Nascimento, and M. Kumar. An improved deep learning-based optimal object detection system from images. https://doi.org/10.1007/s11042-023-16736-5 , 2023.

[6] D. Zhang, Y. Chen, and Z. Li. An effective approach of vehicle detection using deep learning. https://doi.org/10.1155/2022/2019257, 2022.

[7] A. Ammar, A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira. Vehicle detection from aerial images using deep learning: A comparative study. http://dx.doi.org/10.3390/electronics10070820, March 2021.

[8] J. a. Kim, J.-Y. Sung, and S. h. Park. Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition. https://ieeexplore.ieee.org/document/9277040, 2020.

[9] A. Ammar, A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira. Aerial images processing for car detection using convolutional neural networks: Comparison between faster r-cnn and yolov3, 2019.

[10] K. M. Krishna, A. Sowmya, D. Jerusha, and D. Susmitha. Comparative study of vehicle detection using ssd and faster rcnn, 2021.

[11] J. a. Kim, J.-Y.Sung, and S. h. Park. Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition, 2020.

[12] M. Liand Z. Zhang, L. Lei, X.Wang, and X. Guo. Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster r-cnn, yolo v3 and ssd, 2020.

[13] L. Tan, T. Huangfu, L. Wu, and W. Chen. Comparison of yolo v3, faster r-cnn, and ssd for real-time pill identification, 2021.

[14] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman. Comparative analysis of deep learning image detection algorithms, 2021.

[15] H. Phaniharam, M. Atmakuri, and A. Shanmukhi. Comparison of various deep learning algorithms used for object detection.

[16] S. M. Alkentar, B. Alsahwa, A. Assalem, and D. Karakolla. Practical comparation of the accuracy and speed of yolo, ssd and faster rcnn for drone detection, 2021.

[17] N. S. Ouf. Leguminous seeds detection based on convolutional neural networks: Comparison of faster r-cnn and yolov4 on a small custom dataset, 2023.

[18] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni. Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3, 2019.

[19] CVproject. Vehicle detection dataset. https://universe.roboflow.com/cvproject-y6bf4/vehicle-detection-gr77r , dec 2022. visited on 2024-02-07.

[20] Dataset Conversion. Visdrone dataset. https://universe.roboflow.com/dataset-conversion-ipkwb/visdrone-uhzsx , aug 2022. visited on 2024-02-07.

[21] Pak Vehicles. Pak vehicles dataset. https://universe.roboflow.com/pak-vehicles/pak-vehicles , may 2024. visited on 2024-05-07.

[22] Roboflow. Roboflow: Optimize your computer vision workflow. https://roboflow.com/, 2024.

[23] G. Jocher, A. Chaurasia, and J. Qiu. Ultralytics yolov8. https://github.com/ultralytics/ultralytics , 2023.

[24] T. Hashmi, N. Ul Haq, M. Fraz, and M. Shahzad. Application of deep learning for weapons detection in surveillance videos, 05 2021.

[25] N. Ul Haq, M. Fraz, T. Hashmi, and M. Shahzad. Orientation aware weapons detection in visual data : A benchmark dataset, 12 2021.

[26] M. Asad, T. Hashmi, and O. Rasheed. Multiplatform surveillance system for weapon detection using yolov5, 01 2023.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. https://arxiv.org/abs/1512.02325 , 2016.

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. https://arxiv.org/abs/1506.01497 , 2016.

[29] N. Ul Haq, T. Hashmi, M. Fraz, and M. Shahzad. Rotation aware object detection model with applications to weapons spotting in surveillance videos, 05 2021.

[30] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. https://arxiv.org/abs/2004.10934, 2020.