



Superior Use of YOLOv8 to Enhance Car License Plates Detection Speed and Accuracy

Tahreer Abdulridha Shyaa^{1*}, Ahmed A. Hashim²

¹ Computer Science Department, The Iraqi Commission for Computers and Informatics, The Informatics Institute for Postgraduate Studies, Baghdad 10053, Iraq

² Media Technology and Communications Engineering, University of Information Technology and Communications, College of Engineering, Baghdad 10011, Iraq

Corresponding Author Email: phd202120683@iips.edu.iq

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ria.380114>

ABSTRACT

Received: 23 August 2023

Revised: 1 December 2023

Accepted: 12 December 2023

Available online: 29 February 2024

Keywords:

license plates, object detection, faster R-CNN, single shot detector (SSD), YOLO, YOLOv8, deep learning

Recent advances in computer vision help machines understand and process visual information. Computer vision is commonly used to recognise car licence plates, improving traffic monitoring, law enforcement, and parking management. The use of deep learning has improved object detection accuracy, robustness, and speed. Each algorithm has its own advantages and disadvantages, and the choice often depends on the specific needs or application, such as the need for speed versus the need for high accuracy. In this paper, the YOLOv8 was proposed as an object detecting algorithm. Faster R-CNN (Region-based Convolutional Neural Networks) and SSD (Single Shot Detector) were used to implement, evaluate, and compare their results with the proposed algorithm. The three object identification algorithms utilized car licence plate information from photos and video using frameworks as testing datasets. The results showed that due to its capacity to propose regions and classify objects simultaneously, YOLOv8 is suitable for real-time computer vision workloads. dataset size and hyperparameter values are thoroughly examined to determine model performance. Two datasets of varying sizes were used to evaluate methods. Indian number plate and Automatic Number Plate Recognition use YOLOv8 to optimise precision and recall with f1 confidence curve values of 0.700 for small datasets and 0.419 for large datasets.

1. INTRODUCTION

Learning algorithms can provide a finer understanding of the way learning may be challenging within diverse situations [1, 2]. There has been a significant pioneer effort that focuses mostly on suitable software tools for implementing deep learning approaches for picture categorization and object recognition, which provide useful data for semantic analysis of visual media and is associated with a variety of ideas such as image classification, individual behaviour analysis, identifying people by their faces and self-driving cars [3]. The aim of object detection comes to acquire accurate bounding boxes that includes objects of a specific form [4] Object identification can be broken down into two categories: "general object detection" and "detection applications," the former of which aims to study methods of detecting various objects within a unified framework to mimic human vision and cognition, and the latter of which refers to detect in various application scenarios like pedestrian detection, face detection, text detection, and so on [5]. In recent years, the automobile industry and numerous academic organizations have developed deep-learning-based techniques to object recognition that have yielded promising results. Faster R-CNN, and SSD (2D object identification), AVOD (3D object identification) are some of the most extensively used network

designs [6]. Automatic car license plate recognition is one of the main applications based on an object detection method that has received a lot of attention from researchers [7]. It has evolved one of the most critical systems for traffic monitoring and vehicle surveillance. The steps for the car license plate detect process can be illustrated in Figure 1 that includes data gathering, training with deep learning techniques, and ultimately testing the model with an input image to detect the plate [8]:

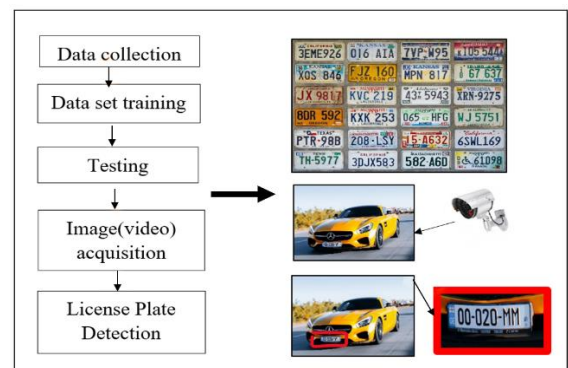


Figure 1. A diagram depicts the essential phases for a license plate detection system

Licence plates vary in size, colour, typeface, and style, making detection difficult. Since licence plates are often photographed against various and complicated backgrounds like vehicles, street signs, and cityscape objects, this is another issue. Finally, lighting, weather, and camera angles affect plate visibility and readability. Deep learning algorithms have been successful in solving these problems due to their various characteristics. Such techniques include Feature Learning, which can identify important licence plate features (such as shape, edges, and character patterns) despite design variations; data augmentation during training to handle environmental conditions; and real-time high processing with accuracy. Integration with recognition and tracking is another capability [9, 10].

This paper presents the use of newly developed promising algorithm (YOLOv8) as object detection technique and compared with two of the highest priority deep learning algorithms that are already in use for object detection R-CNN, and SSD. The license plate will be used as the object being detected. Two different data sets (small and large) will be used to evaluate these algorithms, to demonstrate the effect of the data set size on the results. YOLOv8 which is the latest version of YOLO, and this will be the first work done to employ this version of YOLO in this type of application, YOLOv8 is renowned for its exceptional speed, making it well-suited for real-time detection applications and conducts detection by analysing the entire image, leading to improved contextual understanding and reduction of false positives, beside its capability of detection to numerous object kinds.

2. RELATED WORK

More recently, there has been a great deal of focus on Identification of Vehicle Identification Numbers, which is critical task presents several unique challenges and complexities, so Several Deep learning algorithms have shown competitive performance on a range of datasets proving great success in addressing these challenges an example of this is "Automatic License Plate Detection and Recognition in Thailand" by Ogiuchi et al. [11]. The results from image data studies were encouraging, with a plate identification rate of 94.6% and a character recognition rate of 92.0%. However, the current method is computationally expensive and has a high false positive rate.

In 2017, Li et al. [12] proposed a single forward pass for a deep neural network to recognize license plates and detect text, avoiding intermediate error accumulation and increasing processing speed, The present network is effective and efficient, but there is potential for development especially when it comes to processing speed and supporting a wider range of license plate orientations.

In a study published in 2020, Omar et al. [13] propose a new strategy based on the fusion of various Faster Regions with Convolutional Neural Network (Faster- RCNN) structures. The suggested license plate detection approach employs the AlexNet, VGG16, and VGG19 as three pre-trained CNN models used by the three separate Faster-RCNN modules; the proposed method's accuracy is 97%, Although the existing models work well, a faster R-CNN model could provide the accuracy necessary for real-time tasks while yet maintaining the necessary speed balance.

Salemdeeb and Erturk's 2020 strategy to detecting LPs and identifying their origin, language, and structure is novel [14].

A novel low-complexity convolutional neural network architecture and the YOLOv2 detector with ResNet feature extraction core were used to categorize LPs. The suggested method classifies countries, languages, and designs with 99.57% average detection precision and 99.33% accuracy, However, studies have shown that certain license plates can only be identified at very small sizes. Smaller LPs can still be detected, but with poorer precision, due to the redesign of ResNet50's input layer size.

Zhang et al. [15] attempted to solve the one of the problems associated with license plate detection, based on three traffic video-based (LP) datasets, V-LPDR is a revolutionary unified framework based on deep learning for license plate detection, tracking, and recognition in real-world traffic videos, but it should be notes that Images and videos taken in real life, especially in challenging settings, can sometimes affect the recognition performance of existing LPDR systems, When it comes to low-resolution photos and lighting variation, these systems aren't very good.

SSD-based license plate detection in Iraq will also receive attention in 2021, with segmentation to the plate performed via horizontal and vertical shredding. K-Nearest Neighbors (KNN) was used to narrow down the make and model. Five hundred different vehicles from Iraq were used to test out the new method. The percentage of correctly detected plates is 98%, while the success rate for the segmentation procedure is 96%, Despite the fact that the processing time is rather quick, there are some real-time applications that can require even faster processing [16].

For the first time, as shown in a 2021 paper by Beratoglu and Toreyin [17], LP detection can be accomplished without first fully decompressing the encoded data. Computing costs can be reduced by pre-determining which images will be used for LP recognition. The proposed method makes advantage of HEVC (High Efficiency Video Coding) compressed video sequences. The YOLOv3 Tiny Object Detector is used to spot LPs in computer-made pictures. Additionally, the proposed strategy reduces inference time by more than 30%, YOLOv8 outperformed YOLOv3 in terms of accuracy. YOLO 3 had trouble detecting small objects or objects with overlapping contexts, but YOLOv8 improved its detection accuracy by using more advanced feature extraction and prediction techniques, which allowed it to handle complex scenarios and small objects better.

Chinese Road Plate Dataset (CRPD) is a new dataset proposed in 2022 by Gong et al. [18], which adds multi-objective Chinese LP pictures to the existing public benchmarks. Using the largest publicly available multi-objective Chinese LP dataset with vertices annotations, the images are primarily acquired with electronic surveillance devices, and the network can be trained end-to-end with completely real-time inference efficiency (30 fps with 640p), License plates can be seen in various real-life situations, such as being held in the hand or lying on the ground, as acknowledged in the study. For a complete license plate detection system, it is also valuable and vital to be able to detect and recognize license plates in these different conditions.

In 2022, Srividhya et al. [19] present an algorithm for recognizing the three levels of object boundaries using an inner-outer outline profile (IOOPL), which paves the way for its incorporation into a video surveillance monitoring of traffic system. In intelligent transportation systems (ITS), it is crucial to determine the type of observed items to track safely and

properly predict traffic characteristics. A number plate recognition method based on text correlation and edge dilation techniques was also described.

A paper published in 2023 proposes a solution to the problem of (LP) being confused with road signs, billboards, and other objects in complex environments by changing the conventional method of car-license plate-letter detection into a sequence of (LP) detection, (LP) detection of the four corners where the numbers are located, (LP) correction, and (LP) identification. Thus, a CenterNet object detection algorithm built upon a modified Hourglass-type network is used for more precise detection after the first technique identifies all possible (LP) and feeds them as input parameters. The network size was decreased in half, while accuracy was raised by 6.19 percent on average. The median time for a frame decreased to 2.71 milliseconds (405 frames per second), the method has a problem with efficiently detecting tiny items. To improve detection of small items and eliminate false negatives, the approach needed an increase in the input size. Additionally, while training, the researchers used a number of augmentation techniques, including color processing, centering, distortion, enlargement, reduction, and rotation. These enhancements are useful, but they complicate training and might not be a perfect reflection of the actual world [20].

3. DEEP LEARNING MODEL FOR OBJECT DETECTION OVERVIEW

Even as network architecture evolves, the more common practice is to undertake some tuning based on the original network or use some method to increase network performance [21]. Deep learning's standards of operation center on the process of extracting features from massive datasets via the first layer, before proceeding to alter the representation via hidden layers, ultimately leading to a more abstract and complicated model [22]. Deep learning-based object identification frameworks are broadly classified into two classes, The first is referred to as (one-stage), and it is considered more complex due to the fact that it requires a robust model head to foretell uniform, densely sampled anchors in many orientations, sizes, and aspect ratios [23]. The other way is known as (two-stage), and in this method, object proposals are generated first and then evaluated categorized and regressed [24]. Single Shot Detector (SSD) and You Only Look Once (YOLO) are one-stage methods, whereas Faster R-CNN is a two-stage method. Before applying each strategy to our chosen dataset, we will attempt to clarify the fundamental differences between these three models to provide a thorough understanding of each approach.

3.1 Architecture

The design of an object detection algorithm influences how it analyses incoming data and extracts characteristics for object recognition. The basic network is a condensed version of a typical SSD (Single Shot Detector) architecture with convolutional (conv) feature layers added on top [24]. Scores and offsets for the established anchor boxes are predicted using information from the newest and some of the older base network layers [23]. The shallow or earlier layers help forecast small items, whereas the deep or later layers help detect huge objects [24]. An essential part of any Faster R-CNN (Faster Region-based Convolutional Neural Network) is a) A region

proposal approach that produces "bounding boxes" or locations of likely items in the image. b) Convolutional Neural Network (CNN)-based feature generation for these items. c) A regression layer to fine-tune the bounding box's coordinates d) A classification layer to predict the object's class [25]. Classifying SVMs with R-CNN is time-consuming since it does a ConvNet forward pass for each region proposal instead of sharing computation. Fast RCNN collects features from the entire input image, pools them based on a specified region of interest (RoI), and then feeds the resulting fixed-size features to subsequent fully connected layers that perform classification and bounding-box regression. Features are extracted from the entire image and sent to CNN in batches for labelling and positioning [26]. Joseph Redmon pioneered the YOLO architecture, which addresses detection as a regression problem based on Darknet design. In a single network, YOLO would predict both bounding boxes and class probabilities. The fundamental concept is based on a user-defined size grid cell in which the object is detected if it falls into the cell [27]. YOLOv8, which was launched in January 2023, is anchor-free, reducing the amount of box forecasts and speeding up the non-maximum impression (NMS). YOLOv8 also employs mosaic augmentation during training [28]. Because the YOLO architecture has fewer layers than SSD, it is faster and easier to implement [29].

3.2 Speed

Particularly for real-time uses, the speed of object detection is critical. How many frames per second (fps) each model can handle on the same GPU is a good proxy for this, as shown in Eq. (1). Because they only need to perform a single regression operation on the input image to predict the object image's category and location, one-stage object identification algorithms can find objects quickly [30].

$$FBS = \frac{\text{Number of Frames Processed}}{\text{Time to Process these Frames (in seconds)}} \quad (1)$$

Many studies show that YOLO versions outperform SSDs in terms of speed [31].

The detection speed of the downscaled image is enhanced by a quicker R-CNN method with narrower semantics due to the reduced number of target pixels and the accelerated convolution calculation of the pixels [32]. Figure 2 compares the frames processed per second (FPS) of the Faster RCNN, R-FCN, SSD, and YOLO models [33].

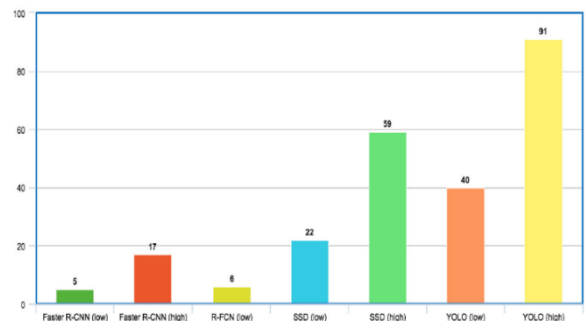


Figure 2. Charts show the no. of frame that been process per second in different deep learning method

The Batch Normalization (BN) has played an important role in deep neural network development [34]. Batch

Normalisation enables us to use much higher learning rates while being less concerned about initialization. It also acts as a regularizer, avoiding the need for Dropout in some cases, hence adopting the Batch Normalisation approach will result in 14 times less training time for the same level of accuracy [35]. Batch Normalization in Faster R-CNN and YOLO help to stabilize the distribution of features in convolutional layers, resulting in faster convergence and higher detection accuracy. SSD [36], for example, can achieve excellent detection accuracy with less training rounds by Normalizing the features at each layer.

3.3 Accuracy

Accuracy is critical in object detection since it influences how successfully an algorithm detects items in a picture. One-stage object detection algorithms perform better in real-time but have lower accuracy, whereas two-stage algorithms perform better in real-time but have lower precision [37]. old YOLO version makes a considerable the amount of localization errors when compared to Fast R-CNN, according to error analysis [38].

Precision and recall Score are metrics that are frequently utilised for Accuracy measurement.

$$\text{Precision} = \frac{\text{True positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

$$\text{Recall} = \frac{\text{True positives}}{\text{True Positives} + \text{False Positives}} \quad (3)$$

Precision, (Eq. 2) which is defined as the percentage of positive examples that were actually recognised, while the percentage of true positives that the model accurately detected is called recall (Eq. 3). Liu presented the SSD method in 2016, which dramatically enhanced object detection accuracy by implementing systems for simultaneous detection at multiple scales and reference points [37]. SSD's higher accuracy is due to its multi-scale feature extraction, successfully enhances the placement accuracy of the SSD network's target prediction layer by fusing the position-sensitive information provided by low-level detail features with the context information provided by high-level semantic features [39]. However, (SSD) performance on small objects remains poor for two main reasons. The first issue is a lack of background information for detecting small items. Furthermore, the Features used for detecting small objects are derived from shallow features that provide no semantic context [40].

4. METHODOLOGY OF WORK

4.1 Dataset and evaluation metrics

Two datasets were chosen that varied in size for testing the algorithms. The first is (Automatic Number Plate Recognition), which consists of 453 JPEG files annotated with bounding boxes containing license plate numbers. PASCAL VOC annotations are included, with xml annotation file for each image providing image metadata, bounding box details, classes, rotation, and other data. The huge dataset is (Indian license plate). It contains over 8000 annotated photos of license plates, the majority of which have Indian number plates. The dataset's annotation is in text format. two videos were also used for testing. The number of classes that were

detected in our test is one class, the picture size that used (640) and (8) batch.

The datasets utilized in this research was tested under various situations, although the previous approaches were tested on a dataset with normal conditions. This mismatch emphasizes the significance of confirming performance results with a broader and more diverse dataset to determine reliability across scenarios. This research addresses the computational requirements and potential restrictions of scaling the system to accommodate a high traffic flow by picking a large data collection, which is necessary for detecting systems like this.

The evaluation metrics that used to compare the result of each testing on small dataset and large one was the charts to show the loss value, and F1 score which is particularly useful for finding the optimal level of confidence that strikes a compromise between the model's recall and precision.

4.2 Testing the small dataset

For a fair comparison of all algorithms, we chose to train the images in (40000) epoch/iteration for each dataset, with certain algorithms experiencing the (Early Stop) state before the estimated epochs ended, which happened when the model's performance on the validation dataset is the same with comparing it to the prior best performance, and that made the model avoids wasting computational resources on additional training when it isn't boosting its performance on unknown data.

The testing begins with SSD algorithm, Figure 3 displays, using the machine learning visualisation tool TensorBoard, the overall loss as a function of the training phases. Minimising loss was the target, and a falling value shows that the model is getting better during training. You can quit training whenever you want if the loss stops going down.

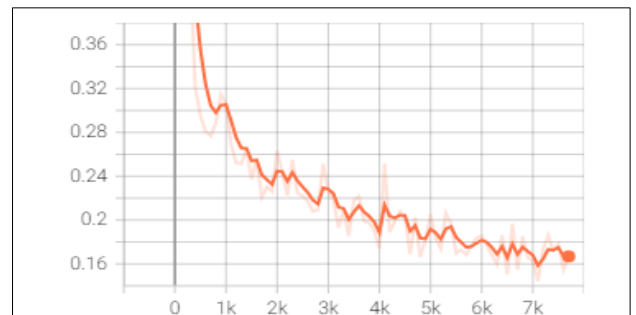


Figure 3. Total loss versus number of steps using SSD

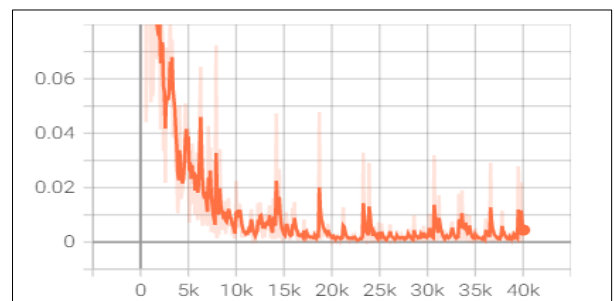


Figure 4. Total loss versus number of steps using Faster RCNN

Figure 4 displays the overall loss as a function of the number

of steps in the TensorBoard training process. The model using the Faster RCNN algorithm performed better than the SSD in this test.

When testing with YOLOv8 the visualization tool for showing the loss was different, because it shows the box_loss, cls_loss and dfl_loss, which represent the overall loss value, see Figure 5.

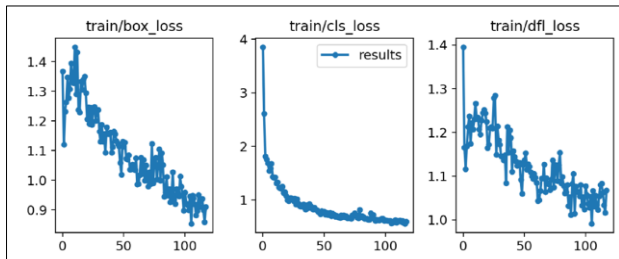


Figure 5. Total loss metrics of testing with YOLOv8

Check out Figure 6 for the f1 confidence curve with YOLOv8. For optimal precision and memory, a confidence value of 0.700 is recommended. A greater level of certainty is preferable in many instances. Since the F1 value seems to be around 0.90, which is not too far from the maximum value of 0.95, it may be optimum to select a confidence of 0.65 for this model. This might be a good design point if we look at the recall and precision figures with a confidence level of 0.70. The precision number remains close to the maximum, while the recall value starts to decline at around 0.95.

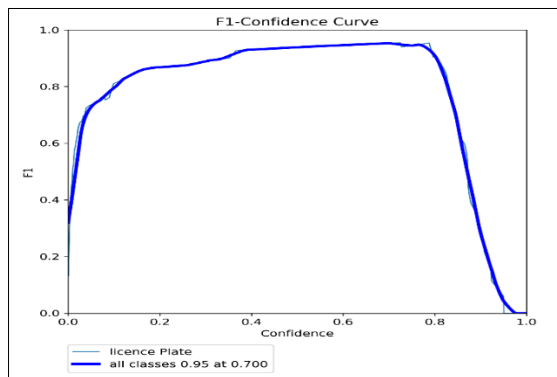


Figure 6. F1 confidence curve with YOLOv8

The test was done on video to mimic the real video application, Figure 7 show the result from using the YOLOv8 algorithm.



Figure 7. The detection on small dataset using YOLOv8

4.3 Testing the large dataset

Training with huge amounts of data in object detection improves generalization, allows for more complicated models, allows for effective data augmentation, and minimizes the risk of overfitting. When we start testing the model after training with a large dataset, we notice that it can detect far Car License Plates that the model trained with small data couldn't detect, even with variations in lighting conditions, backgrounds, poses, and other factors, as well as multiple plate detection. Figure 8 show how the detection improve when training with bigger data set.



Figure 8. The detection on large dataset using YOLOv8

Figures 9-12 demonstrate the results obtained from testing the large dataset utilizing SSD, Faster RCNN and YOLOv8 algorithms respectively; it shown from the assessment metrics a significant change in performance due to the influence of more training on a larger dataset.

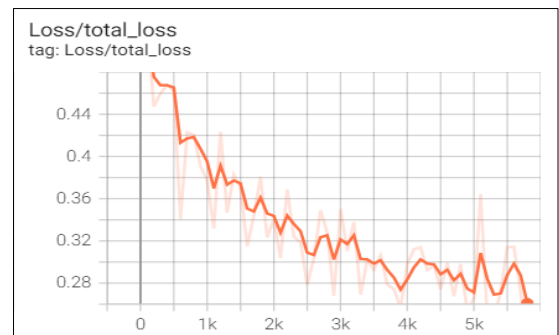


Figure 9. Total loss versus number of steps using SSD

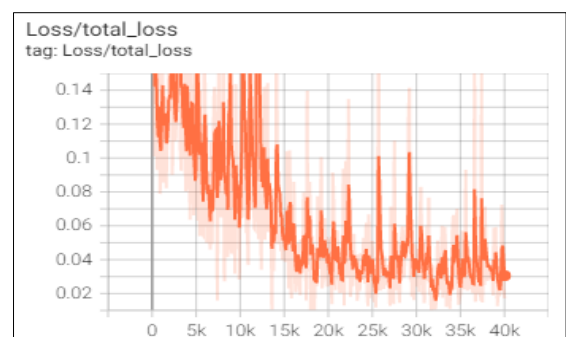


Figure 10. Total loss versus number of steps using Faster RCNN

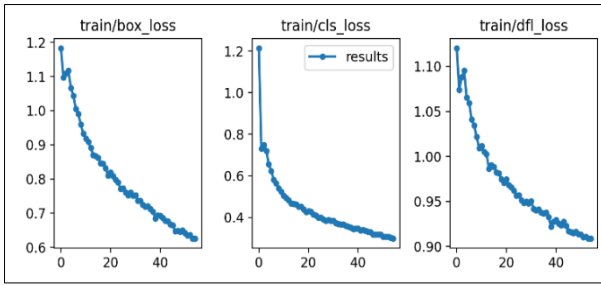


Figure 11. Total loss metrics of testing with YOLOv8

Figure 12 shows the f1 confidence curve with YOLOv8. The confidence value that optimizes the precision and recall is 0.419.

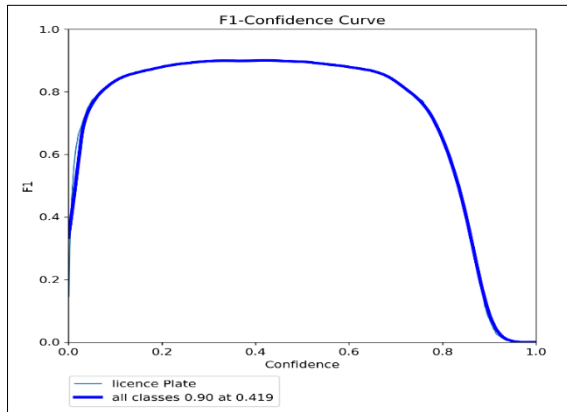


Figure 12. F1 confidence curve with YOLOv8

5. CONCLUSIONS

The main concern for this paper is determining which detector, along with its design, offers the optimal combination of speed and accuracy for a specific application. This study compared a proposed algorithm with two cutting-edge object detection algorithms for car license plate detection. Faster R-CNN, SSD, and YOLOv8, by analysing how well each tactic worked and considering its strengths and shortcomings. All three systems detected license plates accurately and efficiently during the research. There are more sophisticated uses for YOLO than for Faster R-CNN. Because it offers end-to-end training, YOLO makes object detection easier and faster. While both algorithms are capable of producing respectable results, YOLO occasionally surpasses Faster R-CNN in all three metrics. For real-time object detection in images or videos, YOLO is the way to go because it executes single-shot algorithms. Building it is a breeze, and it trains well on complete pictures straight away. When compared to Faster R-CNN, YOLO is a more worthy, quick, and reliable algorithm due to its superior object generalisation. Because of these remarkable benefits, this algorithm is highly recommended and stands out.

REFERENCES

[1] Hashim, A., Rasheed, M., Abdullah, S. (2021). Analysis of bluetooth low energy based indoor localization system using machine learning algorithms. *Journal of Engineering Science and Technology*, 16(4): 2816-2824.

[2] Norouzzadeh, M.S., Morris, D., Beery, S., Joshi, N., Jojic, N., Clune, J. (2021). A deep active learning system for species identification and counting in camera trap images. *Methods in Ecology and Evolution*, 12(1): 150-161. <https://doi.org/10.1111/2041-210X.13504>

[3] Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11): 3212-3232. <https://doi.org/10.1109/TNNLS.2018.2876865>

[4] Roy, S., Unmesh, A., Namboodiri, V.P. (2018). Deep active learning for object detection. In *BMVC*, 362: 91.

[5] Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, 11(3): 257-276. <https://doi.org/10.1109/JPROC.2023.3238524>

[6] Schmidt, S., Rao, Q., Tatsch, J., Knoll, A. (2020). Advanced active learning strategies for object detection. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, USA, pp. 871-876. <https://doi.org/10.1109/IV47402.2020.9304565>

[7] Singh, S. (2021). Automatic car license plate detection system: A review. In *IOP Conference Series: Materials Science and Engineering*, 1116(1): 012198. <https://doi.org/10.1088/1757-899x/1116/1/012198>

[8] Masood, S.Z., Shu, G., Dehghan, A., Ortiz, E.G. (2017). License plate detection and recognition using deeply learned convolutional neural networks. *arXiv preprint arXiv:1703.07330*. <http://arxiv.org/abs/1703.07330>

[9] Shashirangana, J., Padmasiri, H., Meedeniya, D., Perera, C. (2020). Automated license plate recognition: A survey on methods and techniques. *IEEE Access*, 9: 11203-11225. <https://doi.org/10.1109/ACCESS.2020.3047929>

[10] Akpojotor, P., Adetunmbi, A., Alese, B., Oluwatope, A. (2021). Automatic license plate recognition on microprocessors and custom computing platforms: A review. *IET Image Processing*, 15(12): 2717-2735. <https://doi.org/10.1049/ipr.1.2262>

[11] Ogiuchi, Y., Higashikubo, M., Panwai, S., Luenagvilai, E. (2014). Automatic license plate detection and recognition in Thailand. *SEI Technical Review*, 78: 39-43.

[12] Li, H., Wang, P., Shen, C. (2017). Towards end-to-end car license plates detection and recognition with deep neural networks. *arXiv preprint arXiv:1709.08828*. <http://arxiv.org/abs/1709.08828>

[13] Omar, N., Abdulazeez, A.M., Sengur, A., Al-Ali, S.G.S. (2020). Fused faster RCNNs for efficient detection of the license plates. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(2): 974-982. <https://doi.org/10.11591/ijeecs.v19.i2.pp974-982>

[14] Saleemdeen, M., Erturk, S. (2020). Multi-national and multi-language license plate detection using convolutional neural networks. *Engineering, Technology & Applied Science Research*, 10(4): 5979-5985. <https://doi.org/10.48084/etasr.3573>

[15] Zhang, C., Wang, Q., Li, X. (2021). V-LPDR: Towards a unified framework for license plate detection, tracking, and recognition in real-world traffic videos. *Neurocomputing*, 449: 189-206. <https://doi.org/10.1016/j.neucom.2021.03.103>

[16] Abbass, G.Y., Marhoon, A.F. (2021). Iraqi license plate detection and segmentation based on deep learning. *Iraqi Journal for Electrical and Electronic Engineering*, 17(2):

- 102-107. <https://doi.org/10.37917/ijeec.17.2.12>
- [17] Beratoğlu, M.S., Töreyn, B.U. (2021). Vehicle license plate detector in compressed domain. *IEEE Access*, 9: 95087-95096. <https://doi.org/10.1109/ACCESS.2021.3092938>
- [18] Gong, Y., Deng, L., Tao, S., Lu, X., Wu, P., Xie, Z., Ma, Z., Xie, M. (2022). Unified Chinese license plate detection and recognition with high efficiency. *Journal of Visual Communication and Image Representation*, 86: 103541. <https://doi.org/10.1016/j.jvcir.2022.103541>
- [19] Srividhya, S.R., Kavitha, C., Lai, W.C., Mani, V., Khalaf, O.I. (2022). A machine learning algorithm to automate vehicle classification and license plate detection. *Wireless Communications and Mobile Computing*, 2022: 9273233. <https://doi.org/10.1155/2022/9273233>
- [20] Kundrotas, M., Janutėnaitė-Bogdanienė, J., Šešok, D. (2023). Two-step algorithm for license plate identification using deep neural networks. *Applied Sciences*, 13(8): 4902. <https://doi.org/10.3390/app13084902>
- [21] Zhou, X., Gong, W., Fu, W., Du, F. (201). Application of deep learning in object detection. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, Wuhan, China, pp. 631-634. <https://doi.org/10.1109/ICIS.2017.7960069>
- [22] Finjan, R.H., Rasheed, A.S., Murtdha, M., Hashim, A.A. (2021). Arabic handwritten digits recognition based on convolutional neural networks with resnet-34 model. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1): 174-178. <https://doi.org/10.11591/ijeecs.v21.i1.pp174-178>
- [23] Kong, T., Sun, F., Liu, H., Jiang, Y., Shi, J. (2019). Consistent optimization for single-shot object detection. *arXiv preprint arXiv:1901.06563*. <http://arxiv.org/abs/1901.06563>
- [24] Pang, Y., Wang, T., Anwer, R.M., Khan, F.S., Shao, L. (2019). Efficient featurized image pyramid network for single shot detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. Beach, CA, USA, pp. 7336-7344. <https://doi.org/10.1109/CVPR.2019.00751>
- [25] Singh, S., Ahuja, U., Kumar, M., Kumar, K., Sachdeva, M. (2021). Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. *Multimedia Tools and Applications*, 80: 19753-19768. <https://doi.org/10.1007/s11042-021-10711-8>
- [26] Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, 7: 128837-128868. <https://doi.org/10.1109/ACCESS.2019.2939201>
- [27] Bakirman, T. (2023). An assessment of YOLO architectures for oil tank detection from SPOT imagery. *International Journal of Environment and Geoinformatics*, 10(1): 9-15. <https://doi.org/10.30897/ijegeo.1196817>
- [28] Terven, J., Cordova-Esparza, D. (2023). A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv preprint arXiv:2304.00501*. <http://arxiv.org/abs/2304.00501>
- [29] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. <http://arxiv.org/abs/1506.02640>
- [30] Sun, G., Wang, S., Xie, J. (2023). An image object detection model based on mixed attention mechanism optimized YOLOv5. *Electronics*, 12(7): 1515. <https://doi.org/10.3390/electronics12071515>
- [31] Shaghouri, A.A., Alkhatib, R., Berjaoui, S. (2021). Real-time pothole detection using deep learning. *arXiv preprint arXiv:2107.06356*. <https://arxiv.org/abs/2107.06356>
- [32] Qi, L., Li, B., Chen, L., Wang, W., Dong, L., Jia, X., Huang, J., Ge, C.W., Xue, G.M., Wang, D. (2019). Ship target detection algorithm based on improved faster R-CNN. *Electronics*, 8(9): 959. <https://doi.org/10.3390/electronics8090959>
- [33] Sanchez, S.A., Romero, H.J., Morales, A.D. (2020). A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. In *IOP Conference Series: Materials Science and Engineering*, 844(1): 012024. <https://doi.org/10.1088/1757-899X/844/1/012024>
- [34] Yao, Z., Cao, Y., Zheng, S., Huang, G., Lin, S. (2021). Cross-iteration batch normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12331-12340.
- [35] Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448-456. <http://arxiv.org/abs/1502.03167>
- [36] Choi, H.T., Lee, H.J., Kang, H., Yu, S., Park, H.H. (2021). SSD-EMB: An improved SSD using enhanced feature map block for object detection. *Sensors*, 21(8): 2842. <https://doi.org/10.3390/s21082842>
- [37] Redmon, J., Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263-7271. <http://arxiv.org/abs/1612.08242>
- [38] Lian, J., Yin, Y., Li, L., Wang, Z., Zhou, Y. (2021). Small object detection in traffic scenes based on attention feature fusion. *Sensors*, 21(9): 3031. <https://doi.org/10.3390/s21093031>
- [39] Gong, M., Shu, Y. (2020). Real-time detection and motion recognition of human moving objects based on deep learning and multi-scale feature fusion in video. *IEEE Access*, 8: 25811-25822. <https://doi.org/10.1109/ACCESS.2020.2971283>
- [40] Lim, J.S., Astrid, M., Yoon, H.J., Lee, S.I. (2021). Small object detection using context and attention. In *2021 international Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, Jeju Island, Korea (South), pp. 181-186. <https://doi.org/10.1109/ICAIIIC51459.2021.9415217>