Edu Komputika 11 (2) (2024)

Edu Komputika Journal

https://journal.unnes.ac.id/journals/edukom/

Benchmarking YOLOv3 and SSD: A Performance Comparison for Multi-Object Detection

Septian Eko Prasetyo[1] ✉, Chandra Atmaja[2], Muhammad Ardian[2], Alfian Ardhiansyah[1], Ajeng Rahma Sudarni[1], and Mulil Khaira[1]

[1]Informatics and Computer Engineering Education Study Program, Faculty of Engineering, Universitas Negeri Semarang, Indonesia

[2]Department of Electrical Engineering, Faculty of Engineering, Universitas Gadjah Mada, Indonesia

Article Info

Article History:
Received: 2 December 2024
Revised: 22 December 2024
Accepted: 31 December 2024

Keywords:
Object Annotation, Object Detection, PASCAL VOC, SSD, YOLO

Abstract

Multiple object detection remains a significant challenge in the field of computer vision. One of the key factors affecting detection performance is the feature extraction process, especially when objects are relatively small or positioned closely together. This study aims to compare the effectiveness of two popular object detection models, YOLO (You Only Look Once) and Single Shot MultiBox Detector (SSD), in detecting multiple objects within images. These models were selected due to their reported high accuracy and real-time processing capabilities, outperforming traditional methods such as the Hough Transform, Deformable Part-based Models (DPM), and conventional CNN architectures. The models were evaluated using a subset of the PASCAL VOC dataset, which includes object categories such as aircraft, faces, cars, and others, with a total of 1,447 annotated images used in training and testing. The evaluation metric used was mean Average Precision (mAP) to assess detection accuracy. Experimental results indicate that YOLO achieves a mAP of 82.01%, while SSD achieves 70.47%. These findings demonstrate that YOLO provides better performance in detecting multiple objects under the same conditions. Overall, this study confirms the advantages of YOLO in scenarios requiring fast and accurate multi-object detection, highlighting its potential for deployment in real-time applications such as autonomous vehicles, surveillance systems, and robotics. The main contribution of this study lies in providing a comparative performance benchmark between YOLO and SSD on a standard multi-object dataset to guide practical model selection in real-time computer vision tasks.

✉ Correspondence Address:
E11 Bulding, Faculty of Engineering, UNNES,
Semarang, Indonesia, 50229
E-mail: septian@mail.unnes.ac.id

ISSN 2252-6811
E-ISSN 2599-297X

INTRODUCTION

In the current era of rapid technological advancement, computer systems are increasingly being developed to recognize, classify, and interpret objects within digital images—an essential capability in the broader field of computer vision. Multiple object detection, in particular, refers to the task of simultaneously identifying and localizing several instances of predefined object classes within a single image (Li et al., 2018). This process is typically implemented through the use of bounding boxes, which serve to demarcate and annotate each detected object. These boxes not only indicate the presence and position of objects but also assist in isolating unique visual features based on shape, texture, and spatial context (Hendry & Chen, 2019). The ability to detect multiple objects has wide-ranging applications, including autonomous driving, surveillance systems, medical imaging, and smart retail technologies. However, several challenges persist in the detection process. Objects with relatively small sizes are often difficult to detect due to the limited and subtle features they present, which may fall below the resolution threshold of the model. Additionally, scenes that contain multiple objects in close proximity frequently suffer from occlusion and feature overlap, leading to misclassification or false negatives (Li et al., 2018). These challenges necessitate the development of more robust feature extraction techniques and sophisticated detection architectures that can accurately differentiate objects even in densely packed or visually complex environments.

Numerous studies have been conducted to address the challenge of detecting multiple objects in images. One of the earlier methods employed was the Hough Transform, which was used by Barinova et al. (2012) for pedestrian detection. A similar approach was implemented by Leibe et al. (2008) for object segmentation; however, their model struggled to detect certain objects accurately. Another well-known approach is the Deformable Part-based Model (DPM), which has been utilized to detect partially occluded or overlapping objects (Forsyth, 2014). Although DPM performs well in such scenarios, it occasionally produces incorrect bounding boxes in regions where no objects are present.

In more recent years, object detection and classification tasks have increasingly relied on Convolutional Neural Networks (CNNs). For example, Kheradpisheh et al. (2018) proposed a spike-timing-dependent plasticity (STDP) approach for training neural networks, which enables high-speed object detection with low energy consumption—making it suitable for deployment on resource-constrained hardware. Research by Lorencin et al. (2019) applied CNN-based detection to aerial imagery for maritime monitoring and achieved an accuracy exceeding 88%, successfully recognizing marine objects.

Further advancements have enhanced CNN performance. Brahimi et al. (2019) introduced the Boosted Blocks concept, which improved detection efficiency compared to standard CNN architectures. In a subsequent study, Alom et al. (2020) proposed the Inception Recurrent Residual Convolutional Neural Network (IRRCNN), which enhanced accuracy by 4.53%, 4.49%, and 3.56% across different object detection tasks. Additionally, improvements in detection speed have been realized through Fast R-CNN (Girshick, 2015; Wang et al., 2016) and Faster R-CNN (Cao et al., 2019; Ren et al., 2017), which significantly outperform traditional CNNs in terms of processing time while maintaining high accuracy.

A novel approach for multiple object detection, known as YOLO (You Only Look Once), was introduced by Redmon et al. (2016). This model is capable of processing images in real time at a rate of 45 frames per second, offering a significant improvement in speed compared to conventional methods. YOLO has also been reported to outperform traditional models such as Deformable Part-based Models (DPM) and Region-based Convolutional Neural Networks (R-CNN). Due to its high accuracy, YOLO has been successfully applied in various domains. For instance, Sadykova et al. (2020) implemented YOLO to detect outdoor high-voltage insulators using Unmanned Aerial Vehicle (UAV) imagery. Similarly, Ni et al. (2018) employed Light YOLO to recognize ten distinct hand gestures, achieving improvements in model accuracy, processing speed, and overall model size compared to alternative approaches. As of today, YOLO has evolved to its third version (YOLOv3), which includes various updates aimed at enhancing both accuracy and speed (Redmon & Farhadi, 2018).

In parallel, another well-known object detection method, the Single Shot MultiBox Detector (SSD), was proposed by Liu et al. (2016). SSD demonstrated competitive accuracy and speed, surpassing earlier models such as R-CNN. Further improvements to SSD have been explored in subsequent studies. For example, Shi et al. (2019) introduced the Feature Fusion and Enhancement SSD (FFESSD), which improves the extraction of shallow features and delivers better performance than conventional SSD models. Additionally, a research study incorporated a feature fusion structure into SSD, resulting in a 0.8% performance gain over the classic SSD implementation (Jia et al., 2019).

Building upon the various approaches discussed previously, this study aims to conduct a comparative analysis between the YOLO (You Only Look Once) model and the Single Shot MultiBox Detector (SSD) to evaluate their performance and accuracy in multi-object detection tasks. These two models were selected due to their demonstrated superiority in detection accuracy and processing speed compared to conventional methods such as the Hough Transform, Deformable Part-based Model (DPM), and traditional Convolutional Neural Network (CNN) architectures. To assess their reliability, both models were tested using a distinct dataset designed to contain multiple object categories. The goal is to determine the extent to which each model can effectively detect and classify objects under the same experimental conditions.

Despite the growing adoption of YOLO and SSD in object detection tasks, existing literature lacks a direct and comprehensive comparison of their performance under identical conditions—particularly in multi-object detection scenarios. Many studies tend to focus on a single model's enhancement or are limited to specific application domains, leaving a gap in understanding how these two models perform relative to each other in a general context. This gap can hinder informed decision-making when selecting the appropriate model for real-time applications that demand both high accuracy and low latency, such as autonomous navigation, surveillance, or industrial automation.

Therefore, this study aims to conduct a comparative analysis between the YOLO (You Only Look Once) model and the Single Shot Object Detection (SSD) to evaluate their performance and accuracy in multi-object detection tasks. By applying both models to the same dataset and evaluating them using standardized metrics, this research seeks to provide insights into their relative strengths and limitations, offering valuable guidance for future implementations in real-time computer vision systems.

RESEARCH METHODS

This section outlines the research methodology, including the dataset, model architecture, and evaluation procedures for the comparative study. The dataset employed in this research was utilized for both training and testing phases. The proposed models comprise a series of processes ranging from data augmentation to object classification. In this study, the objects are categorized into four classes: airplanes, faces, cars, and miscellaneous objects. The performance of the YOLO and SSD models is compared based on their object detection accuracy and evaluated using the PASCAL VOC metric. A detailed overview of the experimental workflow is illustrated in Figure 1.

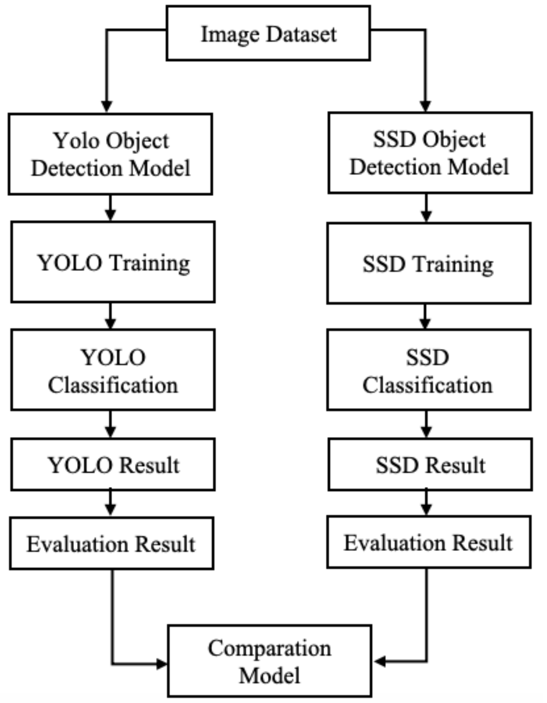


Figure 1. Research step

A. Image Dataset

In this study, we utilized a publicly available dataset from Dataturks, which contains a total of 3,290 annotated object instances intended for both training and testing purposes. The dataset includes four object categories: aircraft, cars, faces, and a miscellaneous "others" class. Each instance is annotated with bounding boxes, making it suitable for supervised learning in object detection tasks. Although limited in the number of object classes, the dataset provides a representative sample of common object detection scenarios with varied object sizes, occlusions, and spatial distributions. For experimental purposes, the dataset was split into 2,632 instances (80%) for training and 658 instances (20%) for testing.

To ensure compatibility with the object detection models employed, we performed a data preprocessing step in which the original annotations were converted into two distinct formats: the YOLO annotation format for the YOLOv3 model and the Lightning Memory-

Mapped Database (LMDB) format for the SSD model. This conversion process was crucial to maintain consistency in data structure across both models, thereby allowing a fair and controlled comparative analysis. Additionally, this preprocessing step enables each model to fully leverage its respective data ingestion mechanism, optimizing training efficiency and model performance.

B. You Only Look Once (YOLO) Model

The YOLO (You Only Look Once) algorithm performs object detection and classification through a streamlined series of steps, as illustrated in Figure 2. Initially, the input image is resized to a fixed resolution of 448 × 448 pixels to ensure consistency across the detection pipeline. The system then applies a single convolutional neural network (CNN) across the entire image, enabling end-to-end processing and prediction in one pass. Unlike traditional methods that involve separate stages for region proposal and classification, YOLO integrates these tasks into a unified model, significantly improving detection speed. The resized image is partitioned into an S × S grid, where each grid cell is responsible for detecting objects whose centers fall within that cell. Each cell predicts a set of bounding boxes along with associated confidence scores. These confidence scores reflect the model's certainty that a box contains an object and the accuracy of the bounding box, as measured by the Intersection over Union (IoU) with the ground truth. Every bounding box prediction includes five parameters: (x, y, w, h, and confidence), where (x, y) represent the center coordinates of the box relative to the grid cell, (w, h) denote the width and height of the predicted object, and the confidence score encapsulates the probability and the IoU. This architecture allows YOLO to achieve real-time object detection with a balance between speed and accuracy, making it well-suited for applications requiring rapid visual recognition.

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{object}) * IOU_{pred}^{truth} = \Pr(\text{Class}_i) * IOU_{pred}^{truth} \quad (1)$$

After obtaining the values of x, y (center coordinates), w (width), h (height), and the confidence score from the detection model, the subsequent step involves calculating the probability that a specific grid cell contains an object. This probability estimation is performed using Equation (1), which integrates the objectness score with the class confidence scores. To facilitate class differentiation during the visualization process, each detected class is assigned a unique label and color code. This approach not only improves interpretability but also enables quick assessment of detection results across multiple object categories. The entire process, including the mapping of predicted bounding boxes and class-specific color assignments, is illustrated comprehensively in Figure 3.

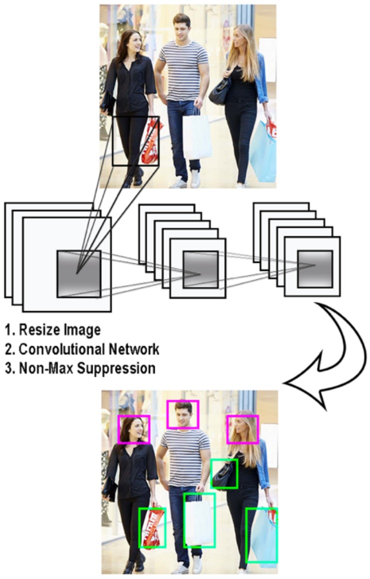


Figure 2. Yolo detection system

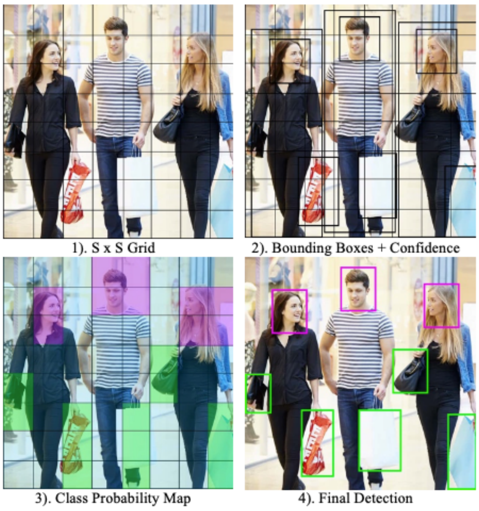
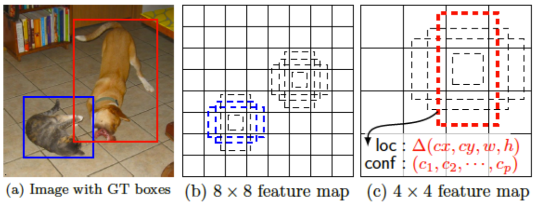


Figure 3. The YOLO model detection

C. Single-Shot Object Detection (SSD) Model

The Single Shot Object Detection (SSD) approach shares fundamental similarities with the YOLO model in its object detection mechanism. Both models follow a pipeline involving the generation of bounding box hypotheses, re-sampling of those boxes, and applying high-confidence classification to localize and identify objects. In SSD, the model only requires input images and corresponding ground truth boxes for training, enabling end-to-end learning for object detection. In this study, SSD utilizes four default bounding boxes with varying aspect ratios, specifically with feature map scales of 8×8 and 4×4. Each default box serves as a prior for predicting the presence, category, and spatial dimensions of potential objects within the image.


loc : $\Delta(cx, cy, w, h)$
conf : $(c_1, c_2, \cdots, c_p)$

(a) Image with GT boxes (b) 8 × 8 feature map (c) 4 × 4 feature map

Figure 4. Loc and confidence bounding box (Liu et al., 2016)

For each default box, the system generates predictions that include the confidence score for each object category, as well as four localization parameters: *cx*, *cy*, *w*, and *h*. Here, *cx* and *cy* represent the center coordinates of the predicted object, while *w* and *h* denote its width and height, respectively. This prediction process is repeated for every default box across the image, leading to multiple object proposals. The scoring and localization process is illustrated in Figure 4. For instance, in Figure 4(a), two objects are identified within the image using the default boxes. These boxes undergo convolutional operations, with each feature map (e.g., 8×8 in Figure 4(b) and 4×4 in Figure 4(c)) applying distinct aspect ratios at different spatial locations. The SSD model then performs confidence computation across all predefined object categories for each box, enabling efficient and accurate multi-object detection.

Unlike YOLO, which resizes input images to 416 × 416 pixels, Single Shot Object Detections (SSD) resize input images to 300 × 300 pixels, allowing for faster computation while maintaining a reasonable level of accuracy. SSD is built upon a feed-forward convolutional neural network architecture that directly predicts a fixed set of bounding boxes and associated class scores in a single pass through the network. The base layer typically adopts a pre-trained image classification network (such as VGG-16) to extract high-level image features. On top of this backbone, SSD integrates additional components, including multi-scale feature maps and convolutional predictors, to enhance detection performance across various object sizes. The complete network architecture of SSD is illustrated in Figure 6.

SSD differs from traditional object detectors that rely on region proposals, such as R-CNN and its variants. While R-CNN follows a two-stage approach—first generating region proposals and then classifying them—SSD implements a one-stage detection strategy similar to YOLO. This approach eliminates the need for a separate proposal generation step, thereby accelerating the detection process. During training, SSD employs a multi-box strategy to match predicted bounding boxes with ground truth, enabling it to handle multiple object classes simultaneously. Default boxes with varying scales and aspect ratios are applied across different feature maps, improving their ability to detect objects of different sizes and shapes. This method has been supported and refined in previous studies, including the work of Sermanet et al. (2013) and He et al. (2014), who emphasized the importance of scale diversity and efficient feature utilization in object detection models.

D. YOLO and SSD Comparison

YOLO (You Only Look Once) and Single Shot Object Detection (SSD) are both one-stage object detection models designed for real-time applications. YOLO performs detection by applying a single neural network to the entire image, dividing it into grids, and directly predicting bounding boxes and class probabilities, resulting in fast inference speed and strong performance on small and overlapping objects. In contrast, SSD relies on a set of default anchor boxes at different aspect ratios and scales, applied across multiple feature maps to detect objects. While SSD generally offers competitive speed, it may exhibit lower accuracy, especially for small objects or when objects are closely packed. These architectural differences influence their detection performance and make them suitable for different deployment scenarios.

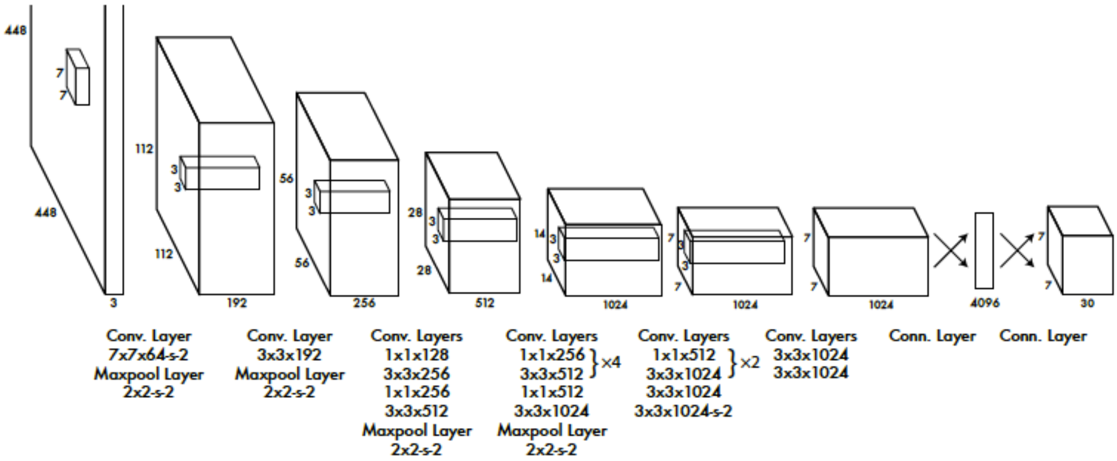


Figure 5. YOLO network architecture (Redmon et al., 2016)

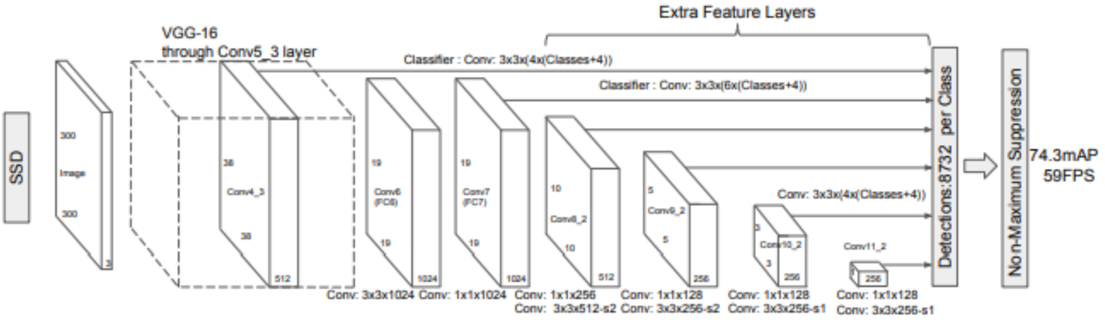


Figure 6. SSD network architecture (Liu et al., 2016)

Figure 5 illustrates the YOLO network architecture, which is based on a single convolutional neural network that divides the input image into an S×S grid and predicts bounding boxes and class probabilities directly from full images in one evaluation. YOLOv3, in particular, utilizes Darknet-53 as its backbone, incorporating residual connections and upsampling to improve detection of small objects. Its design allows for real-time object detection with high speed, making it highly suitable for time-sensitive applications. Figure 6 presents the SSD architecture, which enhances a base network (e.g., VGG-16) by adding several convolutional layers of decreasing size. These layers allow SSD to detect objects at multiple scales and aspect ratios. Unlike YOLO, which performs detection in a single pass over the entire image, SSD leverages multiple feature maps to achieve better accuracy in localizing smaller and variably scaled objects. However, it typically operates at a slower inference speed compared to YOLO.

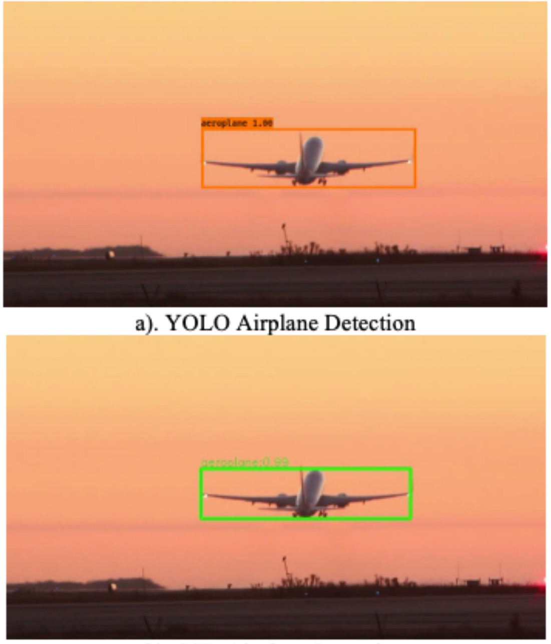
E. Training Configuration

To evaluate the performance of both object detection models, YOLOv3 and SSD were trained using the preprocessed dataset. For the YOLOv3 model, training was conducted with a learning rate of 0.001, a batch size of 16, and a total of 50 epochs using the Adam optimizer, which is well-suited for handling sparse gradients and non-stationary objectives. In contrast, the SSD model was trained using the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9, which is commonly adopted in convolutional architectures to accelerate convergence. The same learning rate and batch size were applied to ensure fairness in comparison.

RESULT AND DISCUSSION

This section presents the results of the comparative analysis between the YOLO and SSD object detection models. Both models were trained using publicly available datasets sourced from Dataturks, with object classes categorized into four main groups: airplane, car, face, and others. The objective of this experiment is to evaluate the accuracy and reliability of each model in detecting multiple objects within a single frame.

The experimental results reveal that YOLO and SSD exhibit different performance levels across the object categories. Overall, both models are capable of detecting objects from multiple classes; however, there are noticeable differences in detection confidence and consistency. Among all the object categories, the airplane class consistently achieved the highest detection accuracy in both models. For instance, as illustrated in Figure 7, YOLO demonstrates a perfect confidence score of 100% in detecting an airplane object (Figure 7a), whereas SSD registers a slightly lower confidence level of 99% on the same image (Figure 7b). Despite this similarity in one example, further evaluation indicates that SSD tends to show a higher degree of variability in confidence levels across different images.



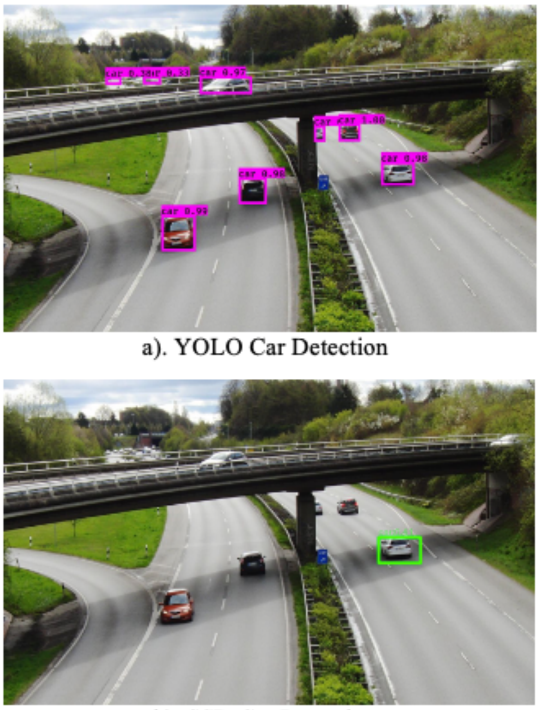
a). YOLO Airplane Detection

b). SSD Airplane Detection

Figure 7. Aeroplan detection result

More critically, SSD occasionally fails to detect certain airplane instances altogether, indicating potential limitations in handling specific object scales or orientations. In contrast, YOLO exhibits a more stable performance, especially in scenarios involving well-defined object boundaries. These findings underscore YOLO's robustness and suitability for applications that demand high detection reliability, particularly in aerial or transportation-related contexts.

Although several object classes achieve high confidence scores, the detection performance for the car class remains relatively low. This limitation is primarily due to image complexity and the spatial proximity of multiple objects within a single frame, which complicates feature extraction. Furthermore, objects located at a distance tend to appear smaller in size, making it more challenging for deep learning algorithms to extract distinct and discriminative features. As a result, detection accuracy for such objects decreases significantly. Figure 8 illustrates the detection performance for the car class. As shown in Figure 8(a), YOLO is able to detect car objects more accurately, including those that are small in size and partially occluded. In contrast, SSD struggles to identify similar car objects, as depicted in Figure 8(b), with noticeably lower confidence scores and several missed detections. Additionally, the bounding boxes generated by SSD often fail to align properly with the object contours, further reducing the model's confidence and accuracy. This issue is not isolated to the car class alone; similar detection inconsistencies and confidence level reductions were observed across other images in the dataset. These findings suggest that YOLO demonstrates greater robustness in handling complex scenes and detecting small or densely clustered objects compared to SSD.



a). YOLO Car Detection

b). SSD Car Detection

Figure 8. Car detection result

To evaluate the performance of both detection models, we employed the PASCAL VOC 2012 evaluation metrics, a widely recognized benchmark in object recognition and detection tasks (Everingham et al., 2010). This evaluation focuses on calculating precision and recall, which are fundamental indicators in assessing the effectiveness of object detection systems. Precision is defined as the ratio of true positive detections (TP) to the total number of positive detections (TP + FP), as formulated in Equation (2), while recall is the ratio of true positive detections to the total number of actual objects (TP + FN), as shown in Equation (3). A True Positive (TP) refers to a correctly identified object with an Intersection over Union (IoU) score equal to or exceeding a predefined threshold. Conversely, a False Positive (FP) is a detection with an IoU below the threshold, and a False Negative (FN) indicates a missed ground truth object.

Since the 2010 revision of the PASCAL VOC guidelines, the average precision (AP) calculation has shifted from a continuous interpolation to an 11-point interpolated average, which is now the default in the official VOC evaluation code (Everingham et al., 2010). This method captures the shape of the precision-recall (P-R) curve by computing the maximum precision at eleven equally spaced recall levels. The formula for calculating AP is provided in Equation (4).

The computed AP values for each object category are presented in Table 1, while the mean Average Precision (mAP)—representing the average of the AP scores across all object classes—serves as the final performance metric to compare the overall accuracy of the detection models.

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{all\ detection} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{all\ ground\ truth} \quad (3)$$

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,\ldots,1\}} \rho interp(r) \quad (4)$$

Table 1. PASCAL VOC Evaluation Result

Method	Average Precision Result			
	Aero	Car	Face	Other
YOLO	100	55.27	92.57	80.21
SSD	62.5	85.36	78.52	55.51

The detailed results of the average precision for each object class are presented in Figure 9, providing a clear comparison of how each detection model performs across various object categories. This per-class evaluation is essential to understand the strengths and weaknesses of each model in detecting specific types of objects such as cars, faces, and aircraft. Furthermore, the overall detection performance is summarized using the mean Average Precision (mAP) metric, as illustrated in Figure 10. The mAP serves as a comprehensive indicator that captures the balance between precision and recall across all categories. By comparing these figures, it becomes evident how the models differ not only in general performance but also in their consistency across different object types. This evaluation provides deeper insight into the practical applicability of each model in real-world scenarios involving diverse and complex visual data.

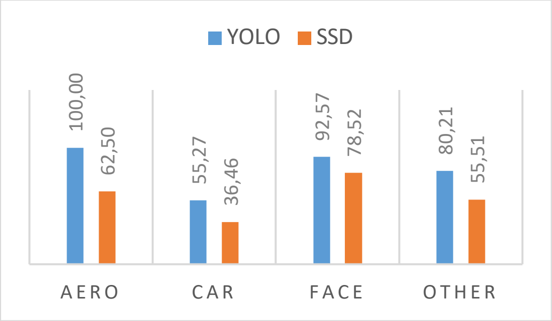


Figure 9. Average precision comparison

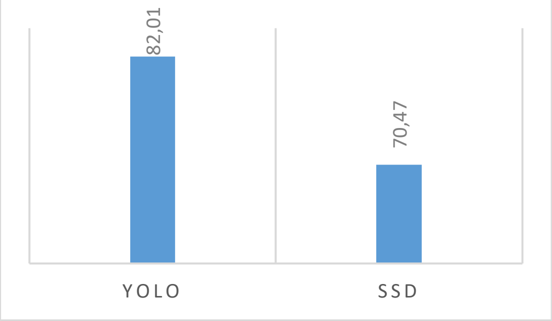


Figure 10. Mean average precision (mAP) comparison

While YOLOv3 demonstrated superior performance in detecting small objects and handling complex scenes, it also exhibited limitations when objects were heavily overlapped or the background was highly cluttered. On the other hand, although SSD showed relatively strong performance for certain classes such as car, its accuracy was inconsistent, especially for smaller or irregularly shaped objects, as reflected in its lower AP scores for the aero, face, and other classes.

In terms of processing speed, YOLO typically offers higher inference rates, reaching up to 45 FPS (frames per second) due to its single-stage architecture that processes the entire image in a single forward pass. This makes it highly suitable for real-time applications. SSD, while also a single-stage detector, generally operates at a slightly lower speed (around 22–30 FPS depending on input resolution), and its performance is more sensitive to object scale and aspect ratio variations. This presents a trade-off: YOLOv3 provides a better balance between speed and accuracy, whereas SSD may be more suitable for resource-constrained systems that do not require high precision.

Overall, the results demonstrate that while both YOLOv3 and SSD are capable of performing object detection tasks effectively, YOLOv3 consistently achieves higher accuracy and faster inference times across the tested categories. These findings confirm the advantages of YOLOv3 in scenarios requiring rapid and precise multi-object detection. Despite SSD being relatively easier to implement and slightly less resource-intensive, it lags behind in performance, especially in real-time contexts. Therefore, YOLOv3 is better suited for applications that demand both speed and accuracy, reinforcing its potential for deployment in practical real-time systems such as autonomous vehicles, surveillance, and robotics.

CONCLUSION

The advancement of multiple object detection in computer vision has progressed rapidly with the introduction of more efficient and faster detection algorithms. While speed is a critical aspect, it must be balanced with detection accuracy and robustness. Several challenges commonly hinder object detection performance, including the presence of multiple objects within a single image, the relatively small size of target objects, close proximity between objects, and inconsistencies or inaccuracies in ground truth annotations. These factors can lead to decreased precision and recall during the detection process. Therefore, the quality of the dataset—both in terms of image resolution and the accuracy of ground truth labeling—plays a crucial role in achieving reliable detection outcomes.

In this study, we conducted a comparative analysis between two prominent object detection algorithms: YOLO and SSD. Both models were evaluated using the same dataset to ensure consistency. The results revealed that YOLO achieved a higher accuracy of 82.01%, while SSD reached an accuracy of 70.47%. These findings suggest that YOLO demonstrates superior capability in handling complex scenes involving multiple objects.

The implications of these results are particularly relevant for real-world applications that require fast and accurate object detection, such as autonomous vehicles, surveillance systems, and robotic vision. In such time-critical environments, YOLO's superior balance between detection speed and accuracy provides a tangible advantage in ensuring timely and reliable system responses.

Future research could explore the performance of these models on more complex and domain-specific datasets, such as those used in medical imaging, traffic analysis, or drone-based aerial surveillance. In addition, further studies should investigate the impact of various augmentation techniques, anchor box optimization, and lightweight model variants to improve detection performance on resource-constrained devices.

REFERENCES

Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., & Asari, V. K. (2020). Improved inception-residual convolutional neural network for object recognition. Neural Computing and Applications, 32(1), 279–293. https://doi.org/10.1007/s00521-018-3627-6

Barinova, O., Lempitsky, V., & Kohli, P. (2012). On detection of multiple object instances using Hough transforms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(9), 1773–1784. https://doi.org/10.1109/TPAMI.2012.79

Brahimi, S., Ben Aoun, N., & Ben Amar, C. (2019). Boosted convolutional neural network for object recognition at a large scale. Neurocomputing, 330, 337–354. https://doi.org/10.1016/j.neucom.2018.11.031

Cao, C., Wang, B., Zhang, W., Zeng, X., Yan, X., Feng, Z., Liu, Y., & Wu, Z. (2019). An improved Faster R-CNN for small object detection. IEEE Access, 7, 106838–106846. https://doi.org/10.1109/ACCESS.2019.2932731

Everingham, M., van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Forsyth, D. (2014). Object detection with discriminatively trained part-based models. Computer, 47(2), 6–7. https://doi.org/10.1109/MC.2014.42

Girshick, R. (2015, December). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 1440–1448). IEEE. https://doi.org/10.1109/ICCV.2015.169

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), Computer Vision – ECCV 2014. Lecture Notes in Computer Science, volume 8691 (pp. 346–361). Springer, Cham. https://doi.org/10.1007/978-3-319-10578-9_23

Hendry, & Chen, R.-C. (2019). Automatic license plate recognition via sliding-window darknet-YOLO deep learning. Image and Vision Computing, 87, 47–56. https://doi.org/10.1016/j.imavis.2019.04.007

Jia, S., Diao, C., Zhang, G., Dun, A., Sun, Y., Li, X., & Zhang, X. (2019). Object detection based on the improved Single Shot MultiBox Detector. Journal of Physics: Conference Series, 1187(4), 042041. https://doi.org/10.1088/1742-6596/1187/4/042041

Redmon, J., & Farhadi, A. (2018, April 8). YOLOv3: An incremental improvement (Tech. Rep.). arXiv. https://doi.org/10.48550/arXiv.1804.02767

Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., & Masquelier, T. (2018). STDP-based spiking deep convolutional neural networks for object recognition. Neural Networks, 99, 56–67. https://doi.org/10.1016/j.neunet.2017.12.005

Leibe, B., Leonardis, A., & Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. International Journal of Computer Vision, 77(1–3), 259–289. https://doi.org/10.1007/s11263-007-0095-3

Li, J., Wong, H.-C., Lo, S.-L., & Xin, Y. (2018). Multiple object detection by a deformable part-based model and an R-CNN. IEEE Signal Processing Letters, 25(2), 288–292. https://doi.org/10.1109/LSP.2017.2789325

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), Computer Vision – ECCV 2016 (Lecture Notes in Computer Science, Vol. 9905, pp. 21–37). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-319-46448-0_2

Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2019). Marine objects recognition using convolutional neural networks. NAŠE MORE, 66(3), 112–119. https://doi.org/10.17818/NM/2019/3.3

Ni, Z., Chen, J., Sang, N., Gao, C., & Liu, L. (2018, October). Light YOLO for high-speed gesture recognition. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 3099–3103). IEEE. https://doi.org/10.1109/ICIP.2018.8451766

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. Paper presented at the 2nd International Conference on Learning Representations (ICLR), Banff, Canada.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779–788). IEEE. https://doi.org/10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Sadykova, D., Pernebayeva, D., Bagheri, M., & James, A. (2020). IN-YOLO: Real-Time detection of outdoor high voltage insulators using UAV imaging. IEEE Transactions on Power Delivery, 35(3), 1599–1601. https://doi.org/10.1109/TPWRD.2019.2944741

Shi, W., Bao, S., & Tan, D. (2019). FFESSD: An accurate and efficient single-shot detector for target detection. Applied Sciences, 9(20), Article 4276. https://doi.org/10.3390/app9204276

Wang, X., Ma, H., & Chen, X. (2016). Salient object detection via fast R-CNN and low-level cues. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 1042–1046). IEEE.