Journal of Robotics and Control (JRC)
Volume 5, Issue 2, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i2.21145

Road Object Detection using SSD-MobileNet Algorithm: Case Study for Real-Time ADAS Applications

Omar Bouazizi [1*], Chaimae Azroumahli [2], Aimad El Mourabit [3], Mustapha Oussouaddi [4]
[1, 3, 4] Systems, and Data Engineering Team, National School of Applied Sciences, Tanger (ENSAT), Abdel Malek Essaadi University, Morocco
[2] Laboratory of Intelligent Systems and Applications (LSIA), Moroccan School of Engineering Sciences (EMSI), Morocco
Email: [1] omar.bouazizi@etu.uae.ac.ma, [2] c.azroumahli@emsi.ma, [3] aelmourabit@uae.ac.ma,
[4] mustapha.oussouaddi@etu.uae.ac.ma
*Corresponding Author

***Abstract*—Object detection has played a crucial role in Advanced Driver Assistance Systems (ADAS) applications, particularly with integrating deep learning techniques. These advancements have improved ADAS applications by enabling more precise object identification, thereby enhancing real-time decision-making. Object detection models can be categorized into two main groups: two-stage and one-stage models. While prior studies reveal that one-stage detectors generally achieve higher frames per second (FPS) at the expense of some accuracy, they remain better suited for real-time ADAS applications. Our study aims to analyze the performance of an object detection model created using SSD-MobileNet, a one-stage detector approach. We focused on identifying road-related objects such as vehicles, and traffic signs. The contribution of our work lies in developing an object detection model using a pre-trained SSD-MobileNet and employing transfer learning. This process involves introducing a new fully connected layer tailored for the specific identification of objects in road scenes. The retraining of the SSD-MobileNet model is executed through GPU-accelerated transfer learning on the MS COCO dataset, incorporating appropriate pre-processing to exclusively include road-related objects. Our findings indicate promising results for the retrained SSD-MobileNet model, achieving an F1 score of 0.801, and a Mean Average Precision (mAP) of 65.41 at 71 FPS. A comparative analysis with other one-stage and two-stage detectors demonstrates the model's performance, surpassing some existing works in the literature related to road object detection. Notably, our model exhibits improved mAP while maintaining a higher FPS, rendering it more apt for ADAS applications.**

Keywords—ADAS; SSD-MobileNet; CNN; Object Detection; Transfer Learning; FPS; MS COCO.

I. INTRODUCTION

Amidst the alarming rise in road accidents, the continuous evolution of ADAS emerges as a solution, showcasing a substantial impact on mitigating these incidents. Lately, the development of ADAS not only aims to enhance driving experiences but also facilitates the creation of autonomous driving [1], [2]. Integrating embedded road object detection solutions plays a pivotal role in achieving ADAS objectives, such as Pedestrian Detection and Traffic Sign Recognition [3], [4]. However, several challenges remain. Specifically, ADAS requires fast object detection systems, leading to a need for real-time processing [5], [6]. Further, balancing computational efficiency and model accuracy is an ongoing challenge, particularly in resource-constrained embedded systems within vehicles [7], [8].

The application of Deep Learning algorithms for Object Detection within ADAS is designed to address these challenges, improving both the accuracy and cost efficiency of detecting road objects [9]. Deep learning utilizes multi-layered neural networks to automatically learn patterns and representations from extensive data, including image analysis. This enhances insights and predictive capabilities across various fields, particularly object detection. In addition, the advancement of technology and the availability of powerful GPU graphic cards have enabled numerous researchers to implement robust deep-learning algorithms for object detection [10].

In literature, Convolutional Neural Networks (CNNs) models, are becoming increasingly chosen as the primary deep learning approach for identifying and categorizing objects within images and video frames [10], [11]. Faster Region Proposal Network (Faster R-CNN) [12], You Only Look Once (YOLO) [13] and Single Shot MultiBox Detector (SDD) [14] stand out as some of the most widely employed CNN models for object detection. These architectures have demonstrated notable success in accurately identifying and localizing objects within images. Their architecture learns and extracts features from large input data through convolutional layers by incorporating a variety of network architectures [15]. Notably, Faster R-CNN [12] used the Region Proposal Network (RPN) to create candidate object frames. It then classified and found items with a sequence of convolutional and fully linked layers. In contrast, YOLO [13] turned the object identification task into a regression problem, using a single model to predict both the object's class and bounding box coordinates. Specifically, object detection approaches based on CNNs can be categorized into two types: two-stage detectors, exemplified by Faster R-CNN, and one-stage detectors, like YOLO. The majority of these detectors employ anchor-based methods. They depend on predefined anchor boxes with diverse scales and aspect ratios, strategically placed across the image. These anchors play a crucial role throughout the training process, aiding the

model in accurately predicting anchor-box coordinates and their associated class probabilities [16].

Creating an object detection model based on CNN entails substantial costs in terms of expertise, computational cost, and the availability of large annotated datasets [17]. The training process involves optimizing various network parameters to minimize classification errors in the output. Consequently, available pre-trained CNNs can be fine-tuned for specific domain object detection tasks using Transfer learning [11]. This enables the model to leverage knowledge learned from specific datasets [9]. The transfer learning approach emerges as a solution for addressing challenges related to object detection including reducing the number of detected classes to enhance the model's performances for a specific field [18]. In literature, many contributions have demonstrated success in the identification of road objects with impressive accuracy using transfer learning. For instance, the study in [19] stands out, employing transfer learning with YOLO architecture to create two object detection models. This work exemplifies the efficacy of leveraging pre-existing knowledge and innovative architectures to achieve robust and accurate results in the field of road object detection and classification.

SSD-MobileNet stands out among existing CNN-based object detection models due to its compelling combination of efficiency and accuracy [20]. The SSD architecture ensures real-time processing. The lightweight design of MobileNet allows for resource-efficient deployment on edge devices with constrained computational capabilities [21], [22]. The inherent multi-scale feature extraction of SSD enhances its ability to detect objects of various sizes within a single pass, contributing to its robust performance [23]. Consequently, the architecture of SSD-MobileNet is particularly suitable for time-sensitive ADAS applications.

The present work has two main objectives. First, review the different object detection models based on the CNN architecture. Second, compare the accuracy and FPS performance of the SSD-MobileNet with existing models. More specifically, an investigation into road object detection is conducted by applying transfer learning to a pre-trained SSD-MobileNet model. This approach is employed to address the challenges of ADAS and facilitate efficient object detection. The research contributions of this work are as follows: (1) Preprocessing and cleaning the MS COCO [24] dataset to include only the road object classes, mainly, vehicles, pedestrians, motorcycles, traffic lights, and stop signs. (2) Retraining a lighter SSD-MobileNet model using transfer learning for road object detection. (3) comparing our SSD-MobileNet model with existing object detection models on two testing datasets, MS COCO, and VOC2007 [25], to understand the model's performance for ADAS applications.

This paper adheres to the following structure: Section II explains how CNNs are used for detecting and classifying objects. It offers a comprehensive review of various object detection models, encompassing both two-stage and one-stage detector architectures. Section III details the preprocessing of the MS COCO dataset used for retraining the SSD-MobileNet model. Section IV provides a summary and discussion of the obtained results. The conclusion integrates the findings and a consideration of potential avenues for future research.

II. CNN FOR OBJECT DETECTION

The use of CNN in creating object detection models has significantly enhanced the accuracy and efficiency of identifying and classifying objects in visual data. One of the earliest CNN architectures is AlexNet, introduced by Krizhevsky et al. [26]. These architectures usually comprised a sequence of convolutional and pooling layers, followed by fully connected layers. Subsequently, novel CNN architectures emerged in the literature, such as ResNets by He et al. [27]. They introduced skip connections, addressing the vanishing gradient problem and enabling the training of profound networks. Further advancements led to the development of more sophisticated CNN architectures, incorporating attention mechanisms. The Transformer architecture, proposed by Vaswani et al. [28], is a notable example of this evolution.

The objective of CNN models developed with these architectures is to automatically learn hierarchical representations of data. This is achieved through the use of convolutional layers, allowing for the extraction of meaningful features at various levels of abstraction [10], [29]. In fact, for the training and evaluation of these models, each input image undergoes a sequence of operations; convolution layers with filters, max pooling, fully connected layers, and activation functions [30]. These processes collectively ascertain the likelihood of an object's presence in an image, assigning probability values within the range of 0 to 1 [31]. This training procedure can extend over several weeks, depending on the used GPU parameters. The training initial step involves configuring the network, determining the number of layers, their dimensions, and the matrix operations linking them [32]. For object detection using CNN, the challenge of overfitting during training is critical. Overfitting occurs when a model performs well on training data but fails to generalize to unseen data [33]. To address this issue, learning optimization techniques play a pivotal role. Regularization methods, such as dropout, have been widely employed to prevent overfitting by randomly disabling neurons during training [33]. Moreover, techniques like data augmentation introduce variations in the training dataset, enhancing the model's ability to generalize [34]. Additionally, optimization algorithms, such as Adam, contribute to efficient convergence and improved generalization performance in object detection tasks [33].

Fig. 1 depicts CNN architecture for object detection. The process of CNN consists of four key components: (1) The input layer. (2) A feature extraction layer containing various convolution and pooling layers. (3) A classification stage that is carried out through Fully Connected layers. (4) The output layers. More precisely, CNN extracts the images' features, where the output of each layer is known as a feature map. The input of the CNN layer is the labeled images, where the output contains the predicted object classes, the bounding boxes, and the confidence score [32], [35].

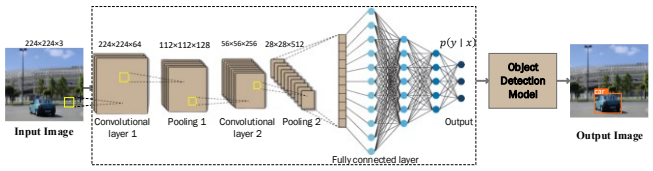

Fig. 1. CNN for object detection

CNN approaches used for object detection predominantly rely on anchor-based methods and can be categorized into two types: Two-stage detectors that are based on region proposal algorithms where they employ a two-step process to localize and classify objects. One-stage detectors where instead of using two detectors simultaneously they use DCNNs without splitting them into separate components [36], [37]. The next subsections provide an overview of the architectures of two-stage and one-stage detectors, followed by a review of diverse architectural approaches used for object detection models.

A. Two-Stage Object Detection Models

A typical two-stage model involves the extraction and classification of region proposals. In this stage, the model generates a set of candidate regions or bounding boxes that are likely to contain wanted objects [38]. After the region proposal, the model integrates a well-established image classification process [39], [40]. Fig. 2 illustrates the operational process of two-stage object detection models. Several two-stage detectors are employed for object detection. The next subsections provide detailed descriptions of each architecture.

Fig. 2. Two-stage model detection diagrams

1) R-CNN architecture: The authors of [41] introduced R-CNN, the first CNN-based model for target detection. R-CNN attained an mAP of 54% on the PASCAL2012 dataset [42]. The main concept behind this region-based approach can be summarized as follows: firstly, an initial segmentation of the input image is performed to generate a substantial pool of regions. These regions are combined recursively to create larger areas, which are then employed to propose potential candidate regions. Lastly, feature extraction and classification are executed through an SVM to predict the coordinates of the target objects [43], [44]. In comparison to traditional detection approaches, R-CNN remarkably enhances accuracy. Nevertheless, this improvement comes at the expense of extensive computational demands. Training R-CNN models demands powerful GPUs, sometimes requiring multiple GPUs or specialized hardware for efficient processing, leading to a lack of computational efficiency. Additionally, scaling the region proposal to a fixed-length feature vector has the potential to distort objects, impacting detection speed, which is limited to 14 FPS [45].

2) SPP-NET architecture: The authors of [46] introduced the Spatial Pyramid Pooling Networks (SPP-Net). This architecture offered a solution to the issue of fixed input size images encountered in R-CNN, which led to reduced detection efficiency. The SPP-Net algorithm concurrently extracts region-specific features from the input image as it progresses through the convolutional layers, performing all convolution calculations. Once the final convolutional layer is traversed, a fully connected layer is added to process the resulting fixed-size feature vector. This model outperforms the R-CNN in terms of both efficiency and speed. SPP-Net attains a mAP of 59.2% on the VOC2007 dataset [16].

3) Fast R-CNN architecture: In 2015, the same authors of [41] introduced the Fast R-CNN architecture in [12]. This architecture builds upon the foundations of SPPNet and R-CNN. Notably, Fast R-CNN diverges from its predecessors by incorporating the softmax function for classification, in contrast to the use of SVM. Additionally, it substitutes the region of interest (RoI) layer with the final pooling layer for convolution. The transformation of the bounding box feature into a fixed-size feature map provides access to the fully connected layer. Ultimately, the last classification layer is substituted with two fully connected (FC) layers. Despite these modifications, Fast R-CNN achieves an impressive accuracy of 70.0% on VOC2007 and VOC2012 [12], [47].

4) Faster R-CNN architecture: Faster R-CNN was created by the authors of [48] in 2015. This novel approach introduces the layer of Region Proposal Network (RPN) to replace the conventional selective search method. This evolved model is divided into two distinct components: a comprehensive CNN block responsible for producing the RPN, and the application of the Fast R-CNN algorithm [49], [50]. Notably, a layer is shared between these two stages, facilitating communication. Remarkably, this algorithm achieves an mAP exceeding 70% on both VOC 2012 and VOC2007 datasets [48].

5) Mask R-CNN architecture: In 2017, the Mask R-CNN algorithm was introduced in [51]. This algorithm is specifically designed for bounding box detection and generates three essential outputs: bounding box coordinates, object masks, and class labels. Through this innovative approach, Mask R-CNN not only generates segmentation masks but also enhances the efficiency of object detection within input images or videos. To facilitate feature extraction, Mask R-CNN adopts ResNet-FPN [52], [53] as its foundational model. It achieved an optimal precision that reaches an mAP of 70.4% on the VoC2017 dataset while maintaining a peak speed of nearly 5 FPS due to extensive computational demands.

B. One-Stage Object Detection Models

In object Detection systems; and when it comes to speed, one-stage models surpass two-stage detectors. Various instances of one-stage model structures can be found, which include YOLO [54], SSD [55], and RetinaNet [56] among others. Fig. 3 illustrates the procedural flow within one-stage models.

Fig. 3. One-stage model detection diagrams

1) YOLO architecture: In 2015, the authors of [13] introduced YOLO, a specialized detection framework designed for real-time predictions. YOLO employs a singular CNN to analyze the entire image, dividing it into distinct regions and then predicting bounding boxes and class probabilities for each region. The confidence score is subsequently established by multiplying the detection probability with the Intersection over the Union (IoU) value. This distinctive approach renders YOLO exceptionally swift, achieving an mAP of over 53% on the VOC2012 dataset [31]. Subsequent improved iterations, YOLOv2 to YOLOv8, attain a processing speed of 45 FPS to 70 FPS [57]. However, YOLO encounters challenges in effectively handling small objects grouped within spatially constrained areas to predict correct and stable Bounding Boxes [58].

2) SSD architecture: In 2016, the authors of [14] introduced the Single Shot MultiBox Detector (SSD) algorithm. Similar to YOLO, this model employs regression and draws inspiration from the use of anchor boxes, aiming to enhance multi-scale object detection performance. SSD combines predictions from multiple feature maps with varying resolutions to handle objects of diverse sizes effectively. The backbone of the SSD algorithm is VGG-16, with the last two fully connected layers being replaced by convolutional layers [59]. SSD achieves an impressive mAP of 76.8% on the VOC 2012 dataset while operating at a speed of 65 FPS [14]. A fundamental contrast between SSD and earlier algorithms lies in their detection approach. More specifically, SSD tests the deeper layers for detection, whereas the older algorithms perform detection tests across multiple distinct layers [55].

3) RetinaNet architecture: Later On, in 2019, the authors of [60] introduced the architecture of RetinaNet. It addresses a challenge in SSD, which encounters class imbalance, resulting in the omission of object classes in certain regions. In contrast to the method of discarding negative samples, RetinaNet employs an innovative loss function referred to as focal loss [61], which effectively attenuates their gradients. This approach applied to the MS COCO dataset, achieved an mAP of 59.1% [60].

III. SSD-MobileNet for Road Object Detection

In this section, we aim to study the applicability of CNN-based object detection models for ADAS applications. Literature shows that one-stage detectors are faster and more suitable for real-time applications at the cost of some accuracy [62], [63]. As a direct result, in this work, we focused on analyzing the performance of the SSD-MobileNet, a one-stage detector model, for road object detection. After, we compare its performance with other existing object detection models from the literature. The original SSD-MobileNet model, designed for 90 different classes, is impractical for ADAS due to computational constraints and real-time processing requirements in ADAS applications. In our study, we optimized the model by reducing the number of classes, reducing complexity, and enhancing suitability. This adjustment is advantageous, as it improves accuracy and FPS when using a GPU.

Our initial step involved preprocessing and refining the MS COCO dataset to exclusively encompass entities relevant to road scenarios, such as vehicles, pedestrians, and traffic signs. Following this, we retrained the SSD-MobileNet using the resulting annotated dataset. The following subsections describe the details of the training dataset, the methodology employed for retraining SSD-MobileNet through transfer learning, and the applied evaluation metrics.

A. The Training Dataset

The construction of an object detection model requires the collection of datasets containing annotated and classified images. For road scenes, several annotated datasets can be used, such as KITTI [64], COCO [25], Berkeley DeepDrive [23], Pascal VOC [25], and the Google Open Image dataset. These datasets can be used for training and evaluation.

For experimental purposes, and to facilitate the training of the SSD-MobileNet model, the MS COCO dataset was chosen. The original MS COCO dataset passed through preprocessing and cleaning to focus on road-related objects, namely cars, persons, motorcycles, traffic lights, and stop signs. These classes were selected to address essential elements in road scenarios, enabling ADAS to comprehend and respond effectively to diverse traffic situations. The car and motorcycle classes play a crucial role in calculating Time to Collision for collision avoidance applications. The person class is vital for enhancing pedestrian safety and preventing accidents. Additionally, classes associated with traffic lights and stop signs are essential for interpreting and responding to traffic signals, ensuring secure navigation at intersections.

After cleaning and preprocessing, the resultant dataset comprises 73k training and validation images extracted from video frames depicting road scenes, each accompanied by corresponding labels. Additionally, the Pascal VOC2007 dataset was employed for testing purposes, undergoing the same preprocessing steps as the MS COCO dataset. Table I provides a summary of the statistical details for the training and the validation datasets used in this study. The training set images are presented sequentially, each paired with a corresponding text file delineating the bounding box coordinates and the respective class of objects within the image.

TABLE I. The Statistics of the Training and the Validation Dataset

Source	Images		Classes	Instances' Statistics	
	Train	*Val*		*Train*	*Val*
MS COCO	64521	8798	Car	232449	41020
			Person	38930	6869
			Motorcycle	14047	2478
			Traffic Light	11493	2028
			Stop Sign	1750	308

B. Methodology

The efficacy of deep learning models is significantly impacted by the dataset's size. In object detection, models trained on limited datasets may encounter pronounced overfitting issues, memorizing training data patterns without generalizing them to new, unseen data [65]. To address this challenge, transfer learning becomes relevant as an approach [66]. Thus, instead of starting training from scratch, we initiate the training process with a pre-trained model (see Fig. 4). More specifically, transfer learning is employed to discard

the fully connected layers of the pre-trained model while retaining the convolutional blocks, and introduce a new fully connected layer. This leads to the fine-tuning of only the top-layer parameters. Consequently, transferring knowledge from a pre-trained model enhances the model's performance and facilitates the utilization of learned features in specific domains, such as road scene object detection. Notably, its effectiveness in handling lighting variations, weather conditions, and diverse road infrastructure, ensures the reliability needed for practical deployment in ADAS applications.

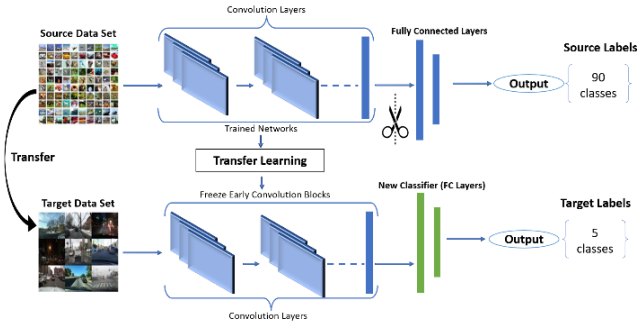

Fig. 4. Transfer learning for road object detection

The outline of our experimentation is depicted in Fig. 5. We set up the SSD-MobileNet [55], [67] for retraining, employing the cleaned and pre-processed dataset. After acquiring the MS COCO dataset, we allocated 15% of the images to a test folder and the remaining 85% to a training folder. Employing Labellmg tools, we annotated the target objects in each image, this tool enabled us to define bounding boxes for missing annotated classes. After, we proceeded to train the object detection model using our GPU graphics card where we specifically used the TensorFlow machine learning library. The experiment was carried out using the NVIDIA GeForce RTX 2060 GPU, 16 of RAM, and 12 CPU.

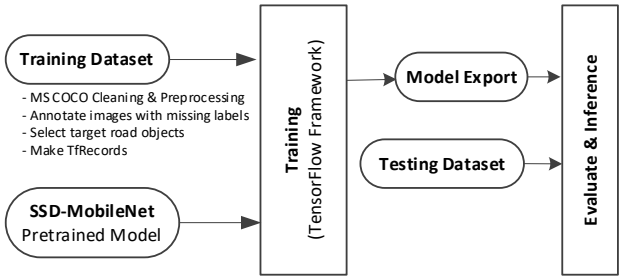

Fig. 5. Object detection with SSD MobileNet

C. *Object Detection Evaluation Metrics*

Within this section, we outline diverse metrics employed for the assessment of neural network models. These evaluation metrics are pivotal tools that enhance our capacity to comprehensively evaluate the SSD-MobileNet models. The evaluation metrics most commonly employed in evaluating object detection models are Precision, Recall, the F1 score, Loss, and mAP [68], [69]. These metrics find widespread application in showcasing the effectiveness of models designed for object detection during and after training.

The Precision, Recall, and F1 scores are denoted by (1), (2), and (3) respectively. These three metrics evaluate the correctness of predictions and the model's capability to detect relevant objects within the training dataset [70]. However, these metrics may not completely capture the complexities of object detection scenarios, particularly in cases involving imbalanced datasets, as is evident in ours (see Table I).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (3)$$

The Cross-Entropy Loss function, as presented by (4), assesses the disparity between predicted and true class probabilities. y_i is the true probability distribution for the class i, $\hat{y}_i$ is the predicted probability distribution, and n is the number of classes. In the training process of our object detection model, this loss function played a pivotal role as a critical optimization criterion. Its primary function was to guide the model in minimizing the difference between its predictions and the actual class distributions, but it cannot be used to reflect on the model's effectiveness in handling real ADAS scenarios [71].

$$H(y, \hat{y}) = -\sum_{i}^{n} y_i . log(\hat{y}_i) \quad (4)$$

We also used the mAP as depicted in (5). AP_i is the average Precision for the class i and n is the number of classes. The mAP consolidated the precision and recall performance across multiple classes. Thus, it provided an assessment of the model's ability to accurately identify and classify objects within the 5 chosen classes [70]. We predominantly relied on mAP to evaluate our model's performance. Considering the distribution of classes in our training dataset (see Table I), mAP provided valuable insights into the model's generalization across different classes, helping assess its suitability for ADAS. For instance, the evaluation of the car class precision is crucial for applications like time to collision, while person and traffic light classes play essential roles in pedestrian safety applications.

$$mAP = \frac{\sum_{i}^{n} AP_i}{n} \quad (5)$$

An additional metric used in evaluating object detection models is Intersection Over Union (IoU). This metric requires two inputs; The predicted coordinates of a detected object within the image, and the actual Bounding Box coordinates of the same object derived from the image's associated label. The IoU produces a result ranging from 0 to 1, representing the precision of object detection [72]. The diagram of Fig. 6 depicts the formula applied for IoU computation. When the IOU value surpasses a specific threshold, a true positive is affirmed; otherwise, it's categorized as a false positive, while false negatives pertain to undetected objects. This metric essentially evaluates the spatial overlap between predicted and true bounding boxes, emphasizing localization accuracy. It holds particular significance for real-time ADAS applications to ensure stable bounding box positioning on a specific object within an image.

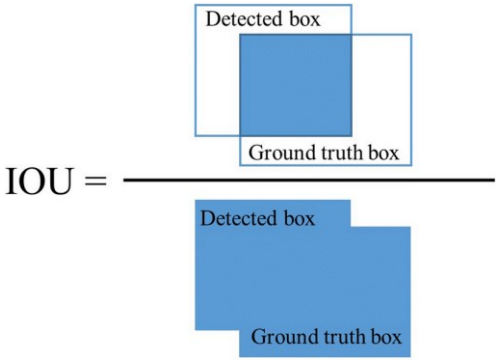

Fig. 6. Computation of the IOU

IV. Results and Discussion

To evaluate the performance of our Object detection model, we computed the F1 score, the mAP, and the FPS on the validation dataset outlined in Section 3.1, which represents 15% of the pre-processed dataset.

Fig. 7 illustrates the training and testing loss curves over 40k steps. Notably, the model's loss stabilizes at around 0.005 after approximately 25k steps. Specifically, this stability suggests that the model is learning the patterns in the training data and approaching an optimal set of parameters. Furthermore, the achieved stability prompted us to stop the training to prevent overfitting. The comparison between the training and validation plots reinforces the idea that the training process is performing optimally.

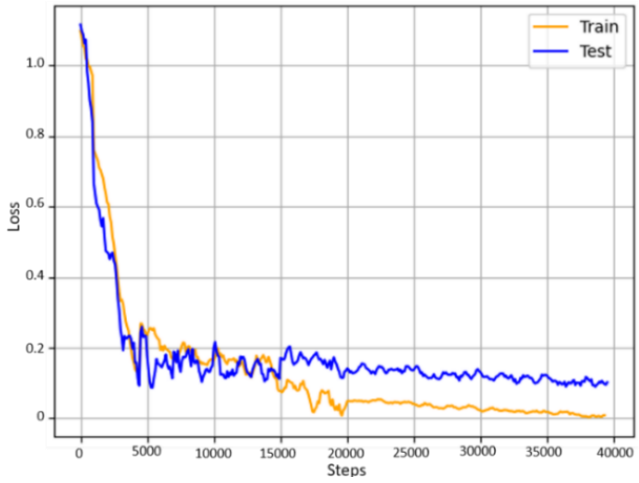

Fig. 7. Train and test loss curve during training

Fig. 8 illustrates the Precision-Recall curve, during training. This curve is commonly employed as an evaluation metric in scenarios characterized by imbalanced class distributions, as is the case in our dataset. Analysis of this curve demonstrates that our model can maintain high precision while capturing a relatively modest number of true positives. This observation is crucial, particularly in the context of ADAS, where our road object detection model must address challenges such as multiple instances with varying object scales and occasional overlap, while the used training dataset contains an imbalanced number of classes.

Fig. 9 illustrates the evolution of Precision, Recall, and F1 scores throughout the training process over 40000 steps. The recall outperforms the other metrics, indicating that the model places a priority on capturing as many true positives as possible rather than false positives. This emphasis on recall is particularly significant for safety-critical ADAS applications, specifically those related to pedestrian safety, where missing a true positive is considered more dangerous than misclassifying a detected object. After training, our model reported an F1 score of 0.801, a precision of 0.756, and a recall of 0.873.

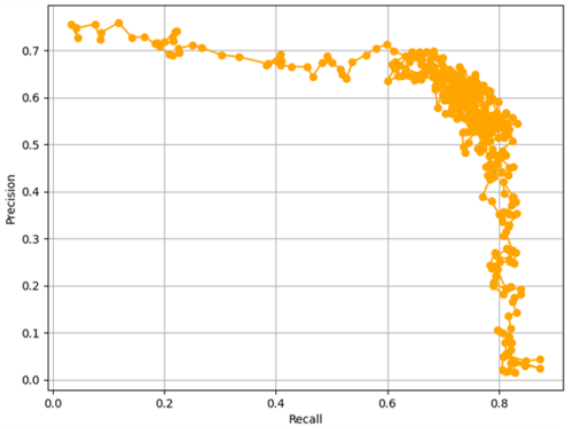

Fig. 8. Precision-Recall Curve during training

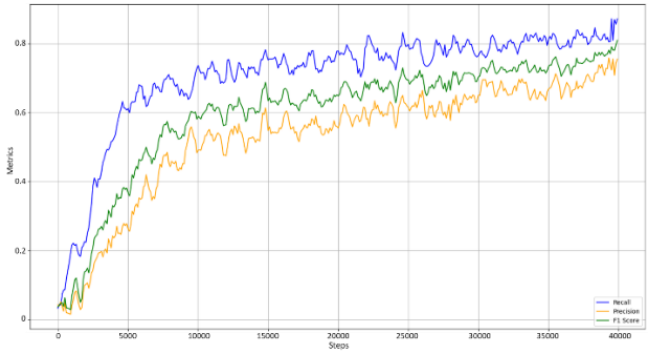

Fig. 9. Precision, Recall, and F1 Score

To further evaluate the performance of the lighter version of SSD-MobileNet with other existing models, we also tested on VOC 2007 after preprocessing, cleaning, and labeling it to only include the selected road objects. In pursuit of a fair comparison, we referenced various studies to include models that were created and evaluated on either the MS COCO or VOC2007 datasets. Moreover, to ensure a consistent metric for FPS, we considered models that were tested on a GPU, similar to our model. In this comparison, we focused on mAP and FPS metrics. The mAP offers a well-balanced evaluation of precision and recall, providing insights into the models' overall performance and localization accuracy. Meanwhile, FPS quantifies the models' efficiency, representing the number of frames processed per second; an essential metric for real-time ADAS applications.

Table II summarizes the mAP results and FPS metrics for the selected models alongside our own. Specifically, [73] introduced a model utilizing a multi-level Feature Pyramid method based on the one-stage architecture of M2det. Another model, detailed in [19], employed the one-stage detector architecture of YOLOv3 and YOLOv4. The last reported mAP score is the score of our model, while the remaining models were described in the review [31]. It is

important to note that in this comparison, certain limitations exist, such as variations in model architecture between one-stage and two-stage detectors. Additionally, for specific models like YOLOv4 as described in [19], the authors implemented a tuning method where model hyperparameters were adjusted for optimal performance.

TABLE II. SSD-MOBILENET PERFORMANCE ON ROAD OBJECT DETECTION

Model	Testing Dataset	@mAP	FPS
M2det [73]	MS COCO	63.96	<5
R-CNN [31]	VOC 2007	66	<5
SPP-Net [31]	VOC 2007	54.2	<5
SDD-VGG16 [31]	VOC 2007	77.2	46
Yolov3 [19]	VOC 2007	76.55	35
Yolov3 – tiny [19]	VOC 2007	62.5	**134**
Yolov4 [19]	VOC 2007	**87.48**	72
Yolov4 [31]	MS COCO	43.5	43.5
SSD-MobileNet (ours)	MS COCO	**65.41**	**71**
SSD-MobileNet (ours)	VOC 2007	**63.46**	**71**

The evaluation of our system yielded a mAP of 65.41, and an FPS of 71, which is notable, particularly considering our use of a relatively smaller training dataset and limited computational resources. Some models achieved a higher accuracy, but they had a lower FPS, making them less suitable for ADAS [19]. Notably, the model created using YOLOv3-tiny achieved the best FPS, however, it required more computational resources for its development, and its mAP was comparatively lower than the other models.

Fig. 10 illustrates the normalized confusion matrix of our SSD-MobileNet model, providing insights into its performance. The high values on the diagonal signify the successful detection of the chosen classes. The class with the lower detection rate is the motorcycle class, primarily due to its limited representation in the training dataset (see Table 1). This performance could be improved by balancing the distribution of classes in the used dataset, thereby preventing any single class from dominating the training data. To address this imbalance, we can apply the strategy of data augmentation before initiating the transfer learning process, emphasizing images containing instances with lower class occurrences.

Fig. 11 depicts multiple instances of inference outcomes generated by our post-trained SSD-MobileNet model. These results not only offer object identification by name but also provide a confidence score, denoting the certainty of the object's presence. These confidence thresholds for detection indicate the model's certainty in its predictions by identifying reliable detections. We set a threshold of 0.6 during inference to decide if a prediction is confident which is useful to balance avoiding false detection and catching real objects, especially in ADAS where safety is crucial. Achieving this balance is crucial for ensuring the reliable performance of road object detection models in real-world scenarios, ultimately enhancing the safety and effectiveness of ADAS applications. The visual representation depicted in Fig. 11 of inference outcomes complements the information revealed by the confusion matrix, collectively painting a comprehensive picture of our model's effectiveness in object detection.

Fig. 10. Confusion matrix normalized

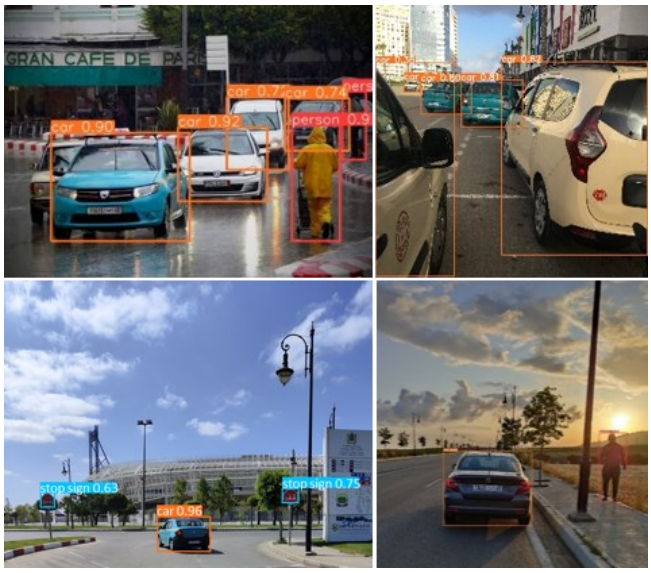

Fig. 11. SSD-MobileNet inference results

V. CONCLUSION

In this paper, we address the challenge of comparing road object detection models fairly, considering factors like performance evaluation, real-time processing, and diverse road environments. More specifically, this study contributes to understanding road object detection models, focusing on implementing SSD-MobileNet in the context of road scenes. Our key contributions are the preprocessing of the MS COCO dataset to include specific road object classes, retraining a lighter SSD-MobileNet model using transfer learning, and conducting a comparative analysis with existing models on MS COCO and VOC2007 datasets. The goal is to assess the model's performance, especially in the context of ADAS applications. Our model demonstrated enhanced performance, achieving an F1 score of 0.801, signifying its ability to detect relevant objects within the training dataset. Additionally, the mAP of 65.41 offers insights into our model's overall performance and localization accuracy. In terms of efficiency for ADAS, the inference was achieved at 71 FPS on a GPU platform, highlighting the model's capability for real-time processing.

Nevertheless, the aspiration of achieving real-time object detection remains a distant goal. Primarily, due to significant challenges posed by computational limitations and inherent complexity of advanced detection models. This paves the way for many prospective trajectories, including optimizing model architectures to deal with the SSD-MobileNet model's architecture limitations and exploring advanced data augmentation methods to deal with the unbalanced road object classes in the MS COCO dataset.

REFERENCES

[1] M. Iqbal, J. C. Han, Z. Q. Zhou, D. Towey, and T. Y. Chen, "Metamorphic testing of Advanced Driver-Assistance System (ADAS) simulation platforms: Lane Keeping Assist System (LKAS) case studies," *Inf. Softw. Technol.*, vol. 155, p. 107104, Mar. 2023, doi: 10.1016/J.INFSOF.2022.107104.

[2] C. A. DeGuzman and B. Donmez, "Drivers don't need to learn all ADAS limitations: A comparison of limitation-focused and responsibility-focused training approaches," *Accid. Anal. Prev.*, vol. 178, p. 106871, Dec. 2022, doi: 10.1016/J.AAP.2022.106871.

[3] T.-M. Guerra, D. Berdjag, P. Polet, and T. A.-T. Nguyen, "Toward a Cooperative ADAS for Train Driving based on Real-Time Human Parameters and Delay Estimation," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 3534–3539, Jan. 2023, doi: 10.1016/j.ifacol.2023.10.1510.

[4] P. M. Greenwood, J. K. Lenneman, and C. L. Baldwin, "Advanced driver assistance systems (ADAS): Demographics, preferred sources of information, and accuracy of ADAS knowledge," *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 86, pp. 131–150, Apr. 2022, doi: 10.1016/j.trf.2021.08.006.

[5] F. Novakazi, J. Orlovska, L. O. Bligård, and C. Wickman, "Stepping over the threshold linking understanding and usage of Automated Driver Assistance Systems (ADAS)," *Transp. Res. Interdiscip. Perspect.*, vol. 8, p. 100252, Nov. 2020, doi: 10.1016/J.TRIP.2020.100252.

[6] I. Athanasiadis, P. Mousouliotis, and L. Petrou, "A Framework of Transfer Learning in Object Detection for Embedded Systems," *arXiv preprint arXiv:1811.04863*, 2018.

[7] I. M. Harms, L. Bingen, and J. Steffens, "Addressing the awareness gap: A combined survey and vehicle registration analysis to assess car owners' usage of ADAS in fleets," *Transp. Res. Part A Policy Pract.*, vol. 134, pp. 65–77, Apr. 2020, doi: 10.1016/j.tra.2020.01.018.

[8] J. Orlovska, F. Novakazi, B. Lars-Ola, M. A. Karlsson, C. Wickman, and R. Söderberg, "Effects of the driving context on the usage of Automated Driver Assistance Systems (ADAS) -Naturalistic Driving Study for ADAS evaluation," *Transp. Res. Interdiscip. Perspect.*, vol. 4, p. 100093, Mar. 2020, doi: 10.1016/J.TRIP.2020.100093.

[9] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, p. 100057, Jul. 2021, doi: 10.1016/j.array.2021.100057.

[10] N. Youssouf, "Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4," *Heliyon*, vol. 8, no. 12, p. e11792, Dec. 2022, doi: 10.1016/j.heliyon.2022.e11792.

[11] L. Geng, J. Sun, Z. Xiao, F. Zhang, and J. Wu, "Combining CNN and MRF for road detection," *Computers and Electrical Engineering*, vol. 70, pp. 895–903, Aug. 2018, doi: 10.1016/j.compeleceng.2017.11.026.

[12] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[14] W. Liu *et al*., "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science*, vol. 9905, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[15] J. Yue *et al*., "Hyperspectral-to-image transform and CNN transfer learning enhancing soybean LCC estimation," *Comput. Electron. Agric.*, vol. 211, Aug. 2023, doi: 10.1016/j.compag.2023.108011.

[16] Y. Song, L. Gao, X. Li, and W. Shen, "A novel robotic grasp detection method based on region proposal networks," *Robot. Comput. Integr. Manuf.*, vol. 65, p. 101963, Oct. 2020, doi: 10.1016/j.rcim.2020.101963.

[17] K. A. Vinodhini and K. R. A. Sidhaarth, "Pothole detection in the bituminous road using CNN with transfer learning," *Measurement: Sensors*, p. 100940, Feb. 2023, doi: 10.1016/j.measen.2023.100940.

[18] A. A. Shetty, N. T. Hegde, A. C. Vaz, and C. R. Srinivasan, "Multi Cost Function Fuzzy Stereo Matching Algorithm for Object Detection and Robot Motion Control," *Journal of Robotics and Control (JRC)*, vol. 4, no. 3, pp. 356–370, Jun. 2023, doi: 10.18196/jrc.v4i3.17041.

[19] S. Wang, "Research towards Yolo-Series Algorithms: Comparison and Analysis of Object Detection Models for Real-Time UAV Applications," in *Journal of Physics: Conference Series*, p. 012021, 2021, doi: 10.1088/1742-6596/1948/1/012021.

[20] D. Biswas, H. Su, C. Wang, A. Stevanovic, and W. Wang, "An automatic traffic density estimation using Single Shot Detection (SSD)and MobileNet-SSD," *Physics and Chemistry of the Earth*, vol. 110, pp. 176–184, Apr. 2019, doi: 10.1016/j.pce.2018.12.001.

[21] I. Shafi, A. Mazahir, A. Fatima, and I. Ashraf, "Internal defects detection and classification in hollow cylindrical surfaces using single shot detection and MobileNet," *Measurement*, vol. 202, p. 111836, Oct. 2022, doi: 10.1016/J.MEASUREMENT.2022.111836.

[22] Y. Liu, Z. Wang, R. Wang, J. Chen, and H. Gao, "Flooding-based MobileNet to identify cucumber diseases from leaf images in natural scenes," *Comput. Electron. Agric.*, vol. 213, p. 108166, Oct. 2023, doi: 10.1016/j.compag.2023.108166.

[23] Z. Chen *et al*., "Fast vehicle detection algorithm in traffic scene based on improved SSD," *Measurement*, vol. 201, p. 111655, Sep. 2022, doi: 10.1016/j.measurement.2022.111655.

[24] K. Tong and Y. Wu, "Rethinking PASCAL-VOC and MS-COCO dataset for small object detection," *J. Vis. Commun. Image Represent*, vol. 93, p. 103830, May 2023, doi: 10.1016/j.jvcir.2023.103830.

[25] S. Chun, W. Kim, S. Park, M. Chang, and S. J. Oh, "ECCV Caption: Correcting False Negatives by Collecting Machine-and-Human-verified Image-Caption Associations for MS-COCO," in *Lecture Notes in Computer Science*, pp. 1–19, 2022, doi: 10.1007/978-3-031-20074-8_1.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2015.

[28] A. Vaswani *et al*., "Attention Is All You Need," *Advances in neural information processing systems*, vol. 30, 2017.

[29] K. Zhang, W. Wang, Z. Lv, Y. Fan, and Y. Song, "Computer vision detection of foreign objects in coal processing using attention CNN," *Eng. Appl. Artif. Intell.*, vol. 102, p. 104242, Jun. 2021, doi: 10.1016/J.ENGAPPAI.2021.104242.

[30] Y. Ji, H. Zhang, Z. Zhang, and M. Liu, "CNN-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances," *Inf. Sci.*, vol. 546, pp. 835–857, Feb. 2021, doi: 10.1016/j.ins.2020.09.003.

[31] Y. Shao, D. Zhang, H. Chu, X. Zhang, and Y. Rao, "A Review of YOLO Object Detection Based on Deep Learning," *Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology*, vol. 44, no. 10, pp. 3697–3708, 2022, doi: 10.11999/JEIT210790.

[32] R. I. H. Abushahma, M. A. M. Ali, O. I. Al-Sanjary, and N. M. Tahir, "Region-based Convolutional Neural Network as Object Detection in Images," in *2019 IEEE 7th Conference on Systems, Process and Control (ICSPC)*, pp. 264–268, 2019, doi: 10.1109/ICSPC47137.2019.9068011.

[33] S. Nandyal and S. Sharanabasappa, "Deep ResNet 18 and enhanced firefly optimization algorithm for on-road vehicle driver drowsiness detection," *Journal of Autonomous Intelligence*, vol. 7, no. 3, Jan. 2024, doi: 10.32629/jai.v7i3.975.

[34] B. P. Kaur *et al*., "An augmentation aided concise CNN based architecture for COVID-19 diagnosis in real time," *Sci. Rep.*, vol. 14, no. 1, p. 1136, Jan. 2024, doi: 10.1038/s41598-024-51317-y.

[35] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, Sep. 2023, doi: 10.1016/j.aiia.2023.07.001.

[36] N. D. Nguyen, T. Do, T. D. Ngo, and D. D. Le, "An Evaluation of Deep Learning Methods for Small Object Detection," *Journal of Electrical and Computer Engineering*, vol. 2020, 2020, doi: 10.1155/2020/3189691.

[37] T. Chen, N. Wang, R. Wang, H. Zhao, and G. Zhang, "One-stage CNN detector-based benthonic organisms detection with limited training dataset," *Neural Networks*, vol. 144, pp. 247–259, Dec. 2021, doi: 10.1016/J.NEUNET.2021.08.014.

[38] P. Song, P. Li, L. Dai, T. Wang, and Z. Chen, "Boosting R-CNN: Reweighting R-CNN samples by RPN's error for underwater object detection," *Neurocomputing*, vol. 530, pp. 150–164, Apr. 2023, doi: 10.1016/J.NEUCOM.2023.01.088.

[39] Y. Huang, Y. Qian, H. Wei, Y. Lu, B. Ling, and Y. Qin, "A survey of deep learning-based object detection methods in crop counting," *Comput. Electron. Agric.*, vol. 215, p. 108425, Dec. 2023, doi: 10.1016/J.COMPAG.2023.108425.

[40] F. M. T. R. Kinasih, C. Machbub, L. Yulianti, and A. S. Rohman, "Two-stage multiple object detection using CNN and correlative filter for accuracy improvement," *Heliyon*, vol. 9, no. 1, p. e12716, Jan. 2023, doi: 10.1016/J.HELIYON.2022.E12716.

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580-587, 2014.

[42] F. Alamri and N. Pugeault, "Improving Object Detection Performance Using Scene Contextual Constraints," *IEEE Trans. Cogn. Dev. Syst.*, vol. 14, no. 4, pp. 1320–1330, Dec. 2022, doi: 10.1109/TCDS.2020.3008213.

[43] M. Zabihi *et al*., "Leveraging Ellipsoid Bounding Shapes and Fast R-CNN for Enlarged Perivascular Spaces Detection and Segmentation," *International Workshop on Machine Learning in Medical Imaging*, pp. 325-334, 2023.

[44] V. K. Sharma and R. N. Mir, "A comprehensive and systematic look up into deep learning based object detection techniques: A review," *Computer Science Review*, vol. 38, p. 100301, 2020, doi: 10.1016/j.cosrev.2020.100301.

[45] A. Umamageswari, S. Deepa, A. Bhagyalakshmi, A. Sangari, and K. Raja, "EmotionFusion: A unified ensemble R-CNN approach for advanced facial emotion analysis," *Journal of Intelligent & Fuzzy Systems*, vol. 45, no. 6, pp. 10141–10155, Dec. 2023, doi: 10.3233/JIFS-233842.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904-1916, 2014.

[47] J. Chen, J. Chen, W. Wang, and Y. Zhu, "Improved model for image tampering monitoring based on fast-RCNN," *2023 2nd International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI)*, pp. 760–764, 2023, doi: 10.1109/ICDACAI59742.2023.00150.

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[49] Y. Wu, "Research on embroidery image recognition based on deep learning," in *Third International Conference on Advanced Algorithms and Signal Image Processing (AASIP 2023)*, p. 61, 2023, doi: 10.1117/12.3005861.

[50] L. Wang, "Faster R-CNN-based pedestrian detection and tracking," in *Third International Conference on Advanced Algorithms and Signal Image Processing (AASIP 2023)*, p. 154, 2023, doi: 10.1117/12.3006095.

[51] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 2, pp. 386–397, Mar. 2020, doi: 10.1109/TPAMI.2018.2844175.

[52] X. Li, T. Lai, S. Wang, Q. Chen, C. Yang, and R. Chen, "Weighted feature pyramid networks for object detection," in *Proceedings - 2019 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking*, pp. 1500–1504, 2019, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217.

[53] P. Chotikunnan, T. Puttasakul, R. Chotikunnan, B. Panomruttanarug, M. Sangworasil, and A. Srisiriwat, "Evaluation of Single and Dual Image Object Detection through Image Segmentation Using ResNet18 in Robotic Vision Applications," *Journal of Robotics and Control (JRC)*, vol. 4, no. 3, pp. 263–277, Apr. 2023, doi: 10.18196/jrc.v4i3.17932.

[54] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," *arXiv preprint arXiv:2305.09972*, 2023.

[55] B. Li, D. Jiang, X. Tang, Y. Sun, and Y. Weng, "Mobilenetv2-SSD Target Detection Method Based on Multi-scale Feature Fusion," *International Conference on Cognitive Systems and Signal Processing*, pp. 206-217, 2023.

[56] R. Del Prete, M. D. Graziano, and A. Renga, "RetinaNet: A deep learning architecture to achieve a robust wake detector in SAR images," in *6th International Forum on Research and Technology for Society and Industry, RTSI 2021*, pp. 171–176, 2021, doi: 10.1109/RTSI50628.2021.9597297.

[57] Y. Li, Y. Ma, and Y. Long, "Protocol for assessing neighborhood physical disorder using the YOLOv8 deep learning model," *STAR Protoc.*, vol. 5, no. 1, p. 102778, Mar. 2024, doi: 10.1016/j.xpro.2023.102778.

[58] S. Bhumbla, D. K. Gupta, and Nisha, "A Review: Object Detection Algorithms," in *ICSCCC 2023 - 3rd International Conference on Secure Cyber Computing and Communications*, pp. 827–832, 2023, doi: 10.1109/ICSCCC58608.2023.10176865.

[59] D. Li, M. Wang, Y. Zhang, and C. Zhai, "Application of an improved VGG and RPN network in precision parts recognition," *Journal of Intelligent & Fuzzy Systems*, vol. 45, no. 6, pp. 9403–9419, Dec. 2023, doi: 10.3233/JIFS-231730.

[60] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 2, pp. 318–327, Aug. 2017, doi: 10.1109/TPAMI.2018.2858826.

[61] X. Wen, X. Yu, Y. Wang, C. Yang, and Y. Sun, "A Hybrid 3D–2D Feature Hierarchy CNN with Focal Loss for Hyperspectral Image Classification," *Remote Sens.*, vol. 15, no. 18, p. 4439, Sep. 2023, doi: 10.3390/rs15184439.

[62] X. Lu, Q. Li, B. Li, and J. Yan, "MimicDet: Bridging the Gap Between One-Stage and Two-Stage Object Detection," in *Lecture Notes in Computer Science*, pp. 541–557, 2020, doi: 10.1007/978-3-030-58568-6_32.

[63] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," in *Journal of Physics: Conference Series, IOP Publishing*, p. 012033, 2020, doi: 10.1088/1742-6596/1544/1/012033.

[64] N. P. A. Duong, A. Almin, L. Lemarié, and B. R. Kiran, "Active Learning with Data Augmentation Under Small vs Large Dataset Regimes for Semantic-KITTI Dataset," in *Communications in Computer and Information Science, Springer Science and Business Media Deutschland GmbH*, pp. 268–280, 2023, doi: 10.1007/978-3-031-45725-8_13.

[65] S. Batool and J. Bang, "Classification of Short Circuit Marks in Electric Fire Case with Transfer Learning and Fine-Tuning the Convolutional Neural Network Models," *Journal of Electrical Engineering and Technology*, vol. 18, no. 6, pp. 4329–4339, Nov. 2023, doi: 10.1007/s42835-023-01490-3.

[66] H. Mzoughi, I. Njeh, M. Ben Slima, and A. BenHamida, "Deep efficient-nets with transfer learning assisted detection of COVID-19 using chest X-ray radiology imaging," *Multimed. Tools Appl.*, vol. 82, no. 25, pp. 39303–39325, Oct. 2023, doi: 10.1007/s11042-023-15097-3.

[67] H. Wang, F. Lu, X. Tong, X. Gao, L. Wang, and Z. Liao, "A model for detecting safety hazards in key electrical sites based on hybrid attention mechanisms and lightweight Mobilenet," *Energy Reports*, vol. 7, pp. 716–724, Nov. 2021, doi: 10.1016/j.egyr.2021.09.200.

[68] M. R. Shoaib, M. R. Elshamy, T. E. Taha, A. S. El-Fishawy, and F. E. Abd El-Samie, "Efficient Brain Tumor Detection Based on Deep Learning Models," in *Journal of Physics: Conference Series*, vol. 2128, no. 1, p. 012012, 2021.

[69] L. Jing and Y. Tian, "Self-Supervised Visual Feature Learning with Deep Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11. pp. 4037–4058, Feb. 16, 2021. doi: 10.1109/TPAMI.2020.2992393.

[70] S. A. Sanchez, H. J. Romero, and A. D. Morales, "A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework," in *IOP Conference Series: Materials Science and Engineering*, 2020, doi: 10.1088/1757-899X/844/1/012024.

[71] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing: A Review Journal*, vol. 126, 2022, doi: 10.1016/j.dsp.2022.103514.

[72] K. Su, L. Cao, B. Zhao, N. Li, D. Wu, and X. Han, "N-IoU: better IoU-based bounding box regression loss for object detection," *Neural Computing and Applications*, vol. 36, no. 6, pp. 3049-3063 2023.

[73] J. Yin, J. Qu, W. Huang, and Q. Chen, "Road damage detection and classification based on multi-level feature pyramids," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 2, pp. 786–799, Feb. 2021, doi: 10.3837/tiis.2021.02.022.