International Journal of Informatics and Computation (IJICOM)
Vol. 5, No.2, 2023
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Improving Vehicle Detection in Challenging Datasets: YOLOv5s and Frozen Layers Analysis

Ahmad Nanda Yuma Rafi[1], Mohamad Yusuf[2]

Abstract

Small datasets and imbalanced classes often cause problems when used as primary research material. In the case of classification and object detection, some researchers proposed Transfer Learning (TF) with several frozen layers. Moreover, YOLO (You Only Look Once) is one of the algorithms that works in real-time object detection. In this research, we focused on evaluating the YOLOv5s version of detecting vehicles in small and imbalanced datasets. The original YOLOv5s were trained and compared with YOLOv5s with the freezing layers method (10 and 24 frozen layers). The experimental results of original YOLOv5s were precision score of 0.779, recall value of 0.933, mAP@0.5 of 0.93 and mAP@0.5:0.95 of 0.684 while YOLOv5s with 10 frozen layers where precision score was decreased to 0.639, but the other value increase with recall value of 0.939, mAP@0.5 of 0.951 and mAP@0.5:0.95 of 0.732. Overall, the version with 10 frozen layers demonstrated superior performance in addressing the challenges of small and imbalanced datasets, particularly excelling in recall and mAP metrics.

Keywords:

YOLOv5s, Image Detection, Transfer Learning, Imbalance Dataset, CNN

1. Introduction

Dealing with small and imbalanced datasets often poses a problem with researchers who conduct research in certain areas where the data is limited and want to avoid building models from scratch [1]. However, it is difficult to obtain large-scale labeled data in most practical applications where a small dataset refers to the lack amount of data, meanwhile an imbalanced dataset refers to the gap amount of data in each class. Even though gaining thousands or millions of data is more recommended the development of technology allows us to do a different approach like applying transfer learning [2].

Deep learning models with transfer learning have been applied in various fields, such as computer vision and natural language processing [1]. This promising technique allows us to reuse the pretrained model from a large dataset that has a broad knowledge of many classes or objects. The reason why this transfer learning works is because the existing knowledge from the network has some intersected class with the new dataset [3]. The fundamental concept is to apply knowledge gained from previous tasks, including data features and model parameters, to support the learning process of a new task. In particular, in image recognition and classification tasks, transfer learning has shown significant benefits, and is getting plenty of attention in the research community [4].

However, we can implement transfer learning on a convolutional neural network using two approaches. First, we freeze the convolutional layers and use the pre-trained model as a feature extractor. The second approach is fine-tuning, whereby we freeze the initial layers and unfreeze deeper convolutional layers. The unfrozen convolutional layers are

Corresponding Author (Mohamad Yusuf: mhd.yusuf@mercubuana.ac.id)
Ahmad Nanda Yuma Rafi[1] (41520010158@student.mercubuana.ac.id)

trained to update the weights. If we have limited new data, we can apply the first approach to prevent overfitting. On the other hand, we can use the second approach with larger datasets to train the deeper layers to detect task-specific features [5].

Furthermore, YOLO algorithms have been widely used in real-time target detection due to their obvious advantages in accuracy and speed. It is an object detector that detects objects in images and localizes them directly into bounding box coordinates and class probabilities [2]. Because of its dependability, validity, speed of detection, and rapidity, it also guarantees real-time identification [4]. Moreover, from the YOLOv1 – v7 version, we decided to use the YOLOv5s version since it has a minimal network size and is more convenient for achieving high-efficiency object detection [6]. Thus, We apply the transfer learning technique using a pre-trained YOLOv5s model on the Common Objects in Context (COCO) dataset to recognize vehicle datasets that include three classes (motorcycle, truck, and car) and compare them with the original model.

2. Related Works

Object detection has become an essential tool for a huge variety of utilization [8]. Since the development of computer hardware and algorithms, now researchers can perform studies about deep learning and neural networks that require high computer specifications. Even though, the development of an algorithm that can compute accurate results in a short time is still needed. Furthermore, to achieve a good detection model, a novel preliminary neural network algorithm was invented. Recently, the most common deep learning-based object recognition models are R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, and YOLO models. The R-CNN family constitutes two-stage object detectors, wherein regions of interest (ROIs) are first extracted, followed by feature extraction and object classification within those ROIs. Consequently, two-stage object detectors generally incur longer detection times compared to one-stage detectors. In contrast, YOLO models belong to the one-stage detector category, directly classifying and regressing candidate boundary boxes without the intermediate step of extracting ROIs, moreover, the YOLOv5 version has been trained on the COCO dataset, which contains 80 classes and more than 200,000 labeled images [2].

Object detection using YOLO was researched by Manikandan et al. where they trained the COCO dataset using the YOLOv5 algorithm to detect several objects including people, vehicles, and animals in various environments. The outcome shows that YOLOv5 achieves good modern performance for a variety of sensing applications, including surveillance, robotics, and autonomous driving [8]. Compared with the other version, YOLOv5 despite being produced by a different author than its predecessors, has higher performance in terms of accuracy and speed among the YOLO family [9]. On the other hand, Arifando et al. proposed The Improved-YOLOv5 model that integrates the GhostConv and C3Ghost Modules into the YOLOv5 network, implements a slim scale detection model, and replaces the SPPF in the backbone with SimSPPF for increased computational efficiency and accurate object detection capabilities. Compared with the other model, The Improved-YOLOv5 model has the best performance in terms of accuracy and inference time. Because of its smaller size, lower memory usage, and faster inference time capabilities, the proposed model was more efficient [10].

Research conducted by Liau et al where compared each YOLOv5 version with and without frozen layers on detecting status recognition system where the YOLOv5s get average inference time (in seconds) 0.0148 (0 frozen layer) and 0.0162 (10 frozen layers) that faster compare among different YOLOv5 models (YOLOv5m, YOLOv5l, YOLOv5x).

Furthermore, Huang et al. proposed a lightweight transfer learning model with pruned and distilled YOLOv5s to identify arc magnet surface defects. The network pruning and knowledge distillation were combined to compress the transferred model. To weaken the loss of accuracy after model compression, a new λ factor was introduced into the confidence loss function of the student model to increase the sensitivity of identifying the defects. Experimental result shows that the model's performance is higher than other regular lightweight models where it could achieve 1.921 MB, and the average inference time was 9.46 ms [6].

On the other hand, Deep learning models with transfer learning have been applied in various fields, such as computer vision and natural language processing [2]. However, Neupane et al. mentioned that there is a greater challenge in training a Deep Learning (DL) model on a custom dataset. DL-based methods have the advantage of accurate detection and classification, but only if the training and test datasets are from the same or similar environment [12]. But, in reality, it is quite challenging to gather a big dataset only in one environment, moreover, the model possibly cannot be implemented in another case. This problem is often called the domain-shift problem where Deep learning is performed under the assumption that the training and test datasets are generated within comparable or identical environments. To minimize this problem, Neupane et al. used the method of transfer learning-based fine-tuning where they successfully improved the accuracy after fine-tuning (71% vs. up to 30%). In this research, we will try to compare the performance of the original YOLOv5s model with the YOLOv5s with the freezing layers method (10 and 24 frozen layers) when applied to an imbalanced dataset [11].

3. Proposed Method

3.1 YOLOv5 Architecture

The original architecture of YOLOv5 is primarily of four components: Input, Backbone, Neck, and Head [10] which consist of several corresponding layers. Input is the component that is used to input the data and normalize the image size to 640 x 640. Then it would choose the 4 random images and process them (cropping, scaling, and layout processing) using mosaic data [10] before it continues to the next processing component. In the backbone, there are 10 total layers and 3 parts (CBS, C3, and SPPF) convolutional layers that are responsible for the learning process. The C3 module is based on the CSPNet [13] concept and includes three standard convolutional layers as well as multiple bottleneck modules. The SPPF module employs a serial max pool to perform multiscale fusion to expand the receptive field's feature map [10].

Meanwhile, the neck will be responsible for fusing the backbone features and improving their representational power using Feature Pyramid Network (FPN) and Pixel Aggregation Network (PAN) structures. The FPN structure conveys semantic information from top to bottom through an upsampling operation, whereas the PAN structure transmits location information from bottom to top via a downsampling operation. Furthermore, the head corresponds to predicting the bounding boxes and class probabilities for each object within the input image.

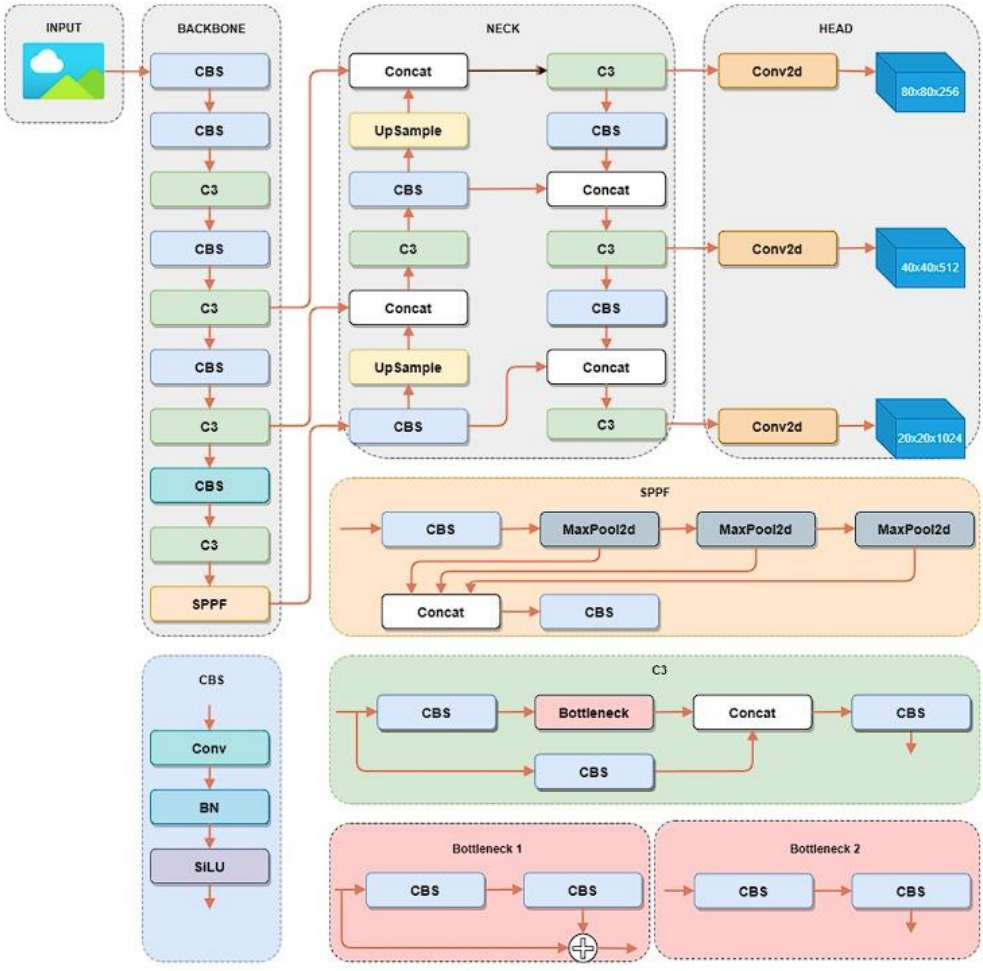

Fig.1 YOLOv5 Structure by Arifando et al

In Fig.1 we can see the connection of each component and its layers. The learning process will divide into three different sizes of the image, starting from the second C3 in the backbone that is directly concated with C3 layers in neck components. This is because in YOLOv5 architecture, originally the algorithm would divide the image into 3 different sizes (as we can see in the head output) to predict the object. In this research, we choose YOLOv5s among the other versions since it is widely used for deployment and real-time inference because of its size and capability.

3.2 Transfer Learning

Dealing with challenging datasets often caused problems, such as overfitting, noise, outliers, and sampling bias. Transfer learning is a machine learning technique where a model trained on one task is adapted to perform a second related task. With this method, we can transfer knowledge from previous tasks to new ones with relatively little training data, rather than learning new tasks from scratch through a large number of data [6] such as COCO or ImageNet, and allows us to utilize a limited amount of training data without experiencing a decline in performance.

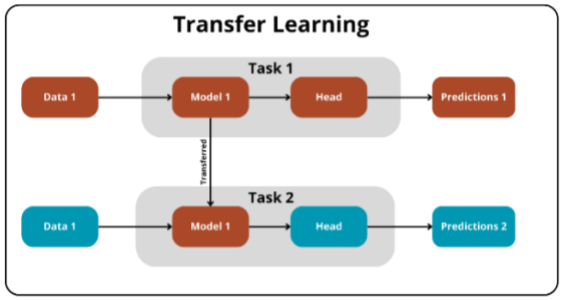

Fig.2 Transfer Learning

In this research, we will try to implement a transfer learning method that uses pre-trained YOLOv5s where the model is already trained on the COCO dataset, which contains 80 classes and more than 200,000 labeled images. The transfer learning technique will be applied through freezing 10 and 24 layers. Furthermore, 10 layers refer to the freezing of the backbone architecture that will keep the weight from the pretrained data and still update the rest of the layer's weight. However, if we freeze the whole layers (24 layers) the model is supposed to work better since we keep all of its weight from the pre-trained data.

3.3 Performance Analysis

In this paper, we adopt mean average precision (mAP) [14] as a performance metric. For each detected class, a precision-recall curve is generated, where recall (r) is defined as the ratio of all positive samples to all samples in that particular class. Precision (p) is a measure of the ratio of correctly identified positive samples to the total number of positive samples in the dataset:

$$r = \frac{tp}{tp+fn}, \qquad p = \frac{tp}{tp+fp}, \qquad (1)$$

Where tp, fn, and fp represent true positive, false negative, and false positive, respectively.

The Average Precision (AP) calculates curves from precision to recall by sampling the curve at all unique recall values or for every true positive. AP can be defined as follows, where M is the number of expected samples:

$$AP = \frac{1}{M} \sum_r p(r) \qquad (2)$$

Where r is the set of numbers obtained through the formula:

$$r = \left\{ \frac{k}{M} \mid 0 \le k \le M, k \in N_0 \right\} \qquad (3)$$

The mAP is the mean value of AP for each class. It can be defined as follows, where N is the number of classes:

$$mAP = \frac{\sum_{i=1}^{N} AP(i)}{N} \qquad (4)$$

By integrating these parameters into our evaluation framework, we will systematically assess the results to gain a comprehensive understanding of the model's performance. The utilization of mean average precision (mAP) as a primary performance metric, along with the examination of precision-recall curves for each identified class, enables a comprehensive measurement of the model's precision across different aspects of the classification assignment.

4. Experimental Setup

4.1 Data Collection

An imbalanced dataset in machine learning occurs when the distribution of classes within the dataset is unequal. Specifically, one class (the minority class) has notably fewer instances than another class (the majority class). This imbalance presents challenges for machine learning algorithms, particularly when the objective is to train a model for accurate prediction or classification of instances from the minority class.

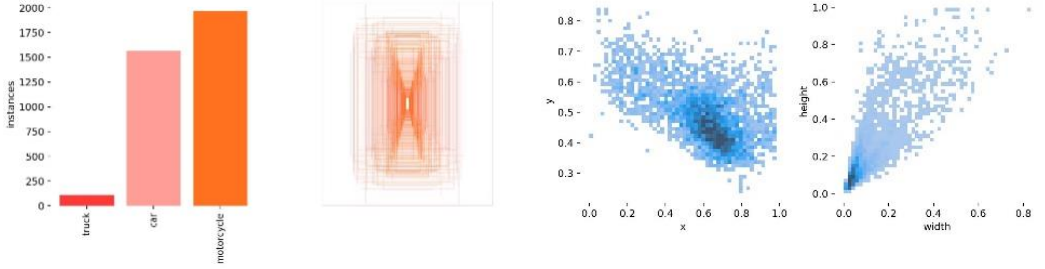

Fig.3 Data Distribution

In this research, the vehicle dataset was collected in West Jakarta, Indonesia using an iPhone 11 camera where we divided the object into three classes (truck, car, and Motorcycle). Furthermore, to gather imbalanced data, we only took 3% of the dataset for the truck class. This method will make sure the data tend to be biased towards the majority class. However, since we want to analyze the transfer learning model, we expect the problem can be solved with the proposed model.

4.2 Model Implementation

In our proposed analysis, we aimed to conduct object detection using an imbalanced dataset. The initial model was trained without freezing any layers, allowing it to learn weights from the respective datasets over 100 epochs with a batch size of 16. Despite the dataset's small size and imbalance, the YOLOv5s pretrained model, unfrozen layers, demonstrated impressive results with a precision of 0.779, a recall of 0.933, and a mAP@50 score of 0.93. Nevertheless, it's important to highlight that the mAP@50-95 metric presented a lower value of 0.684, signaling opportunities for enhancement to encompass a more diverse array of object detection scenarios.

Fig.4 YOLOv5s Backbone and Head algorithm

Additionally, we experimented with freezing all layers of the YOLOv5 model, incorporating the entire convolutional model with pretrained weights. In this approach, we retained the model's architecture and weights, solely adapting the detection class to align with the characteristics of the dataset in use. However, it is noteworthy that the YOLOv5s pretrained model did not perform optimally when all layers were frozen. The results indicated a precision of 0.549, recall of 0.881, mAP@50 of 0.749, and mAP@50-95 of 0.556, suggesting limitations in achieving optimal performance under these conditions.

Table 1 Model evaluation result

Model	Precision	Recall	mAP@50	mAP@50-95
YOLOv5s (0 FL)	**0.779**	0.933	0.93	0.684
YOLOv5s (10 FL)	0.639	**0.939**	**0.951**	**0.732**
YOLOv5s (24 FL)	0.549	0.881	0.749	0.556

On the other hand, we conducted an evaluation using the pretrained YOLOv5s model with 10 frozen layers (FL) on the same dataset. The complete freezing of the backbone in the architecture ensures that the model's weights in this section continue to rely on the pretrained data from YOLOv5s. Remarkably, the recall, mAP@50, and mAP@50-95 metrics demonstrated superior performance in comparison to other models (0 FL and 24 FL), attaining consecutive values of 0.939, 0.951, and 0.732. However, it's crucial to emphasize that the precision result, while commendable, still fell short of the performance achieved by the 0 FL model, registering at 0.639.

5. Result and Analysis

The assessment of YOLOv5s on an imbalanced dataset yielded notable outcomes. Training the model without freezing any layers resulted in exceptional overall performance, indicated by a precision of 0.779, recall of 0.933, and mAP@50 of 0.93. However, the mAP@50-95 metric displayed a relatively lower value of 0.684, suggesting potential areas for improvement in addressing a wider array of object detection scenarios. On the other hand, freezing all layers of the YOLOv5s pretrained model yielded less favorable results, with a precision of 0.549, recall of 0.881, mAP@50 of 0.749, and mAP@50-95 of 0.556. Unexpectedly, performance significantly improved when 10 layers were frozen, achieving the highest recall, mAP@50, and mAP@50-95 values among the evaluated models at 0.939, 0.951, and 0.732, respectively. However, it's important to note that the precision result remained below that of the model with no frozen layers, recording at 0.639.

The study's results emphasize the complex impact of using frozen layers on how well YOLOv5s performs with imbalanced datasets. As a result, we need to understand and manage trade-offs, especially in terms of precision and important metrics. The choice to freeze specific layers brings in a mix of factors that affect how effective the model is overall. By pointing out these detailed dynamics, the research highlights the careful balance needed when optimizing YOLOv5s for scenarios with imbalanced datasets. These insights are helpful for both practitioners and researchers, providing a better understanding of the consequences and considerations when using frozen layers to enhance object detection in challenging dataset conditions.

6. Conclusion

In conclusion, our study provides valuable insights into the impact of frozen layers on the performance of YOLOv5s in the context of handling imbalanced datasets. The results highlight a nuanced interplay of factors, particularly emphasizing trade-offs between precision and essential metrics. Notably, the unexpected improvement in performance observed when 10 layers were frozen suggests a potential avenue for enhancing the model's adaptability to imbalanced dataset scenarios. Our findings contribute to the existing body of knowledge by shedding light on the delicate balance required for optimizing YOLOv5s under these conditions. While our observations confirm the importance of careful consideration when using frozen layers, future work could delve into further understanding the specific mechanisms that lead to the observed improvements.

Furthermore, investigating the transferability of our results to varied datasets and domains has the potential to offer a more comprehensive understanding of the effectiveness of frozen layers in improving object detection performance. This research establishes a foundation for future studies to expand upon, underscoring the continuous advancement and fine-tuning of methodologies within the realms of computer vision and object detection.

Acknowledgment

I extend my gratitude to Mohamad Yusuf, S.Kom., M.C.S, for his invaluable guidance and unwavering support as the supervisor of this research project. His expertise, constructive feedback, and dedication significantly contributed to the successful completion of this study. Additionally, I express my deep appreciation to Universitas Mercu Buana for providing the essential facilities and lectures that enriched my academic journey. The conducive research environment and resources offered by the university played a crucial role in the development and execution of this research.

References

[1] Z. Situ, S. Teng, W. Feng, Q. Zhong, G. Chen, J. Su, and Q. Zhou, “A transfer learning-based YOLO network for sewer defect detection in comparison to classic object detection methods,” *Developments in the Built Environment*, vol. 15, p. 100191, 2023, doi: 10.1016/j.dibe.2023.100191.

[2] Y. Y. Liau and K. Ryu, “Status Recognition Using Pre-Trained YOLOv5 for Sustainable Human-Robot Collaboration (HRC) System in Mold Assembly,” *Sustainability*, vol. 13, no. 21, p. 12044, 2021. [Online]. Available: https://doi.org/10.3390/su132112044.

[3] A. Wiranata, S. A. Wibowo, and R. Patmasari, “Analysis of Transfer Learning on Faster R-CNN for Vehicle Detection,” in *Proc. 2nd Symposium of Future Telecommunication and*

Technologies (SOFTT), 2018. [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/softt/article/view/8440.

[4] A. Magotra and J. Kim, “Improvement of Heterogeneous Transfer Learning Efficiency by Using Hebbian Learning Principle,” *Applied Sciences*, vol. 10, p. 5631, 2020, doi: 10.3390/app10165631.

[5] S. Zakhary and A. Benslimane, “On location-privacy in opportunistic mobile networks, a survey,” *J. Netw. Comput. Appl.*, vol. 103, pp. 157–170, Feb. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804517303557.

[6] Q. Huang, Y. Zhou, T. Yang, K. Yang, L. Cao, and Y. Xia, “A Lightweight Transfer Learning Model with Pruned and Distilled YOLOv5s to Identify Arc Magnet Surface Defects,” *Applied Sciences*, vol. 13, no. 4, p. 2078, 2023, doi: 10.3390/app13042078.

[7] M. U. Khan, M. Dil, M. Misbah, A. Orakazi, M. Alam, and Z. Kaleem, “Translearn-YOLOx: Improved-YOLO with transfer learning for fast and accurate multiclass UAV detection,” 2022.

[8] L. Manikandan, R. Malavika, B. Anjali, and S. Chandana, “OBJECT DETECTION USING YOLO V5,” *Eur. Chem. Bull.*, vol. 12, no. S3, pp. 6226–6233, 2023, doi: 10.31838/ecb/2023.12.s3.698.

[9] T. Diwan, et al., “Object detection using YOLO: challenges, architectural successors, datasets and applications,” Multimedia Tools and Applications, vol. 82, pp. 9243–9275, 2022.

[10] R. Arifando, S. Eto, and C. Wada, “Improved YOLOv5-Based Lightweight Object Detection Algorithm for People with Visual Impairment to Detect Buses,” *Applied Sciences*, vol. 13, no. 9, p. 5802, 2023, doi: 10.3390/app13095802.

[11] B. Neupane, T. Horanont, and J. Aryal, “Real-Time Vehicle Classification and Tracking Using a Transfer Learning-Improved Deep Learning Network,” *Sensors*, vol. 22, no. 10, p. 3813, 2022, doi: 10.3390/s22103813.

[12] B. Neupane, T. Horanont, and J. Aryal, “Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis,” *Remote Sensing*, vol. 13, p. 808, 2021.

[13] C. Y. Wang, H. Y. M. Liao, I. H. Yeh, Y. H. Wu, P. Y. Chen, and J. W. Hsieh, “CSPNet: A New Backbone that can Enhance Learning Capability of CNN,” *arXiv*, 2019.

[14] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.