Semi-supervised learning approach to multiclass object detection with obscure and overlapping boundaries

Lahiru R. Rathnayake[1], and Ruwan D. Nawarathna[2]
[1]Postgraduate Institute of Science, University of Peradeniya, Peradeniya 20400, Sri Lanka
[2]Department of Statistics and Computer Science, University of Peradeniya, Peradeniya 20400, Sri Lanka
email: [1] lahrumesh28@gmail.com, [2] ruwand@pdn.ac.lk

***Abstract*— Multiclass object detection has a variety of uses in the field of computer vision and many of the object detection algorithms are implemented for real-world tasks. A common issue in the area of multiclass object detection is the overlapping objects with unclear boundaries. We propose a method to automatically detect objects with obscure and overlapping boundaries in an image by utilizing a fine-tuned object detection model based on YOLOv3 coupled with a semi-supervised learning approach. To test the model, a dataset containing 5000 images of 8 different classes of grocery items was used. Also, we employed a semi-supervised learning technique called pseudo labeling to minimize the time for labeling of the data. We show that pseudo labeling eliminates the need for expensive manual verification and labeling process which minimize requirements for domain experts in many applications. The final model trained with semi-supervised learning archives a mean average precision(mAP) of 0.895 on the test data set.**

Keywords*— Computer Vision, Multiclass Object Detection, Pseudo Labeling, Semi-supervised learning

I. INTRODUCTION

Over the last years, researchers in the field of Computer Vision have relied heavily upon machine learning approaches that focused on the problem of identifying the visual appearance of objects. From traditional image classification problems [1], they shifted to the more challenging tasks of object detection and segmentation [2]. Neural network based multiclass object detection is one of the prominent applications of Computer Vision[3]. It is concerned with finding and locating specific objects in an image. It appears in many real-world situations, such as medical imaging, traffic management systems, face recognition and self-driving cars. Humans can identify and learn real-world objects effortlessly, but lack of human resources makes it difficult and expensive to obtain human labor for object identification. Identifying objects depend on their appearances which fall into two classes. Some objects may be clearly separated from the other objects as shown in Fig. 1(a), which falls into category of individual isolated object detection. In the other class, objects may overlap with other objects and only a part of the object may be seen as displayed in Fig. 1(b) since the object boundaries are overlapped. This creates a much more challenging object identification problem of overlapped and obscured objects.

In almost all object identification and localization applications, it is required to use multiclass object detection techniques. We decided to develop a multiclass overlapped object detection model for a real-world problem of grocery item detection (See Fig. 1). Most supermarkets rely on traditional barcode readers for item identification during checkout. This process takes a lot of time and can result in long queues and dissatisfied customers. During the period of a pandemic such as Covid-19, dealing with these long queues can be problematic. Multiclass object detection method would make the cashiers' checkout process quicker and easier. On many occasions, customers use a basket or a trolley to gather grocery items. Generally, these items are visibly overlapped with other items as shown in Fig. 1(b) where edges and boundaries are unclear.

From the bucket of Deep Learning algorithms, Convolution Neural Networks (CNNs) are commonly used for image classification and object detection[4] . Edge detection is one of the influential concepts in the field of Computer Vision which is the first phase of object detection [5]. Typically, the first few layers of a CNN perform the feature extraction required for object detection including edge detection. There are two categories of object detection methods. The first one is the two-stage detector in which the model proposes a set of regions of

Figure 1 a) Individual objects non-overlapping object boundaries and b) overlapped objects obscure object boundaries.

interest through select search or regional proposal networks and performs classification on the selective regions [6]. This approach is mainly used in R-CNN, Fast R-CNN and Faster R-CNN models [7], [8], [9]. The second one is the one-stage detector that skips region proposals and run detection directly over a dense sampling of possible locations. Examples for this category include AttentionNet, SSD and YOLO models [10], [11], [12]. Drid et al. [13] presents a comprehensive review of detecting overlapping objects from two-stage and one-stage detector models by using PASCAL VOC dataset [14]. They have proposed a model to combine both models to enhance performance. We propose an algorithm to deal with the challenging task of identification of objects with obscure and overlapping boundaries using a CNN-based

multi-object detection model trained with a semi-supervised algorithm [15].

The remainder of the paper is organized as follows. Section 2 presents the background, details of the models and techniques used, and the related work of the study. The proposed model is described in Section 3 and the experimental results after applying the proposed model on a sample data set are summarized and discussed in Section 4. Finally, Section 5 provides concluding remarks and future works.

II. BACKGROUND

A. Machine Learning for Object Detection

There are distinct machine learning algorithms available for object detection tasks. In general, machine learning algorithms can be divided into 3 main types, namely, supervised learning, unsupervised learning, and semi-supervised learning[16]. The majority of the problems in machine learning use supervised learning methods. Supervised learning models in computer vision problems learn from the labeled dataset which contains input images attached with appropriate labels. Unsupervised learning algorithms work with unlabeled data. Semi-supervised learning which is used in this study uses a combination of supervised and unsupervised learning techniques because it works with both labeled and unlabeled data.

Bounding box is one of the widely used image annotation methods for object detection in machine learning[17]. The purpose of adding bounding boxes is to highlight the visible contents of the image as shown in Fig. 2.

Figure 2 Bounding box annotations

The bounding box is a rectangular box that is determined by a point, width, and height according to the pixels in an image. In object detection methods input is an image with one or more objects and output is one or more bounding boxes and class label for each bounding box. Image annotation extends to instance segmentation where object boundaries are highlighted at pixels level. In supervised learning, human input is required to annotate enormous amounts of data manually which can be extremely challenging. Usually, deep learning models require considerably large data sets to make the final model more accurate and robust. It would be such a waste if unlabeled data is not used for creating the object detection model due to the tedious labeling process. To overcome this issue, we propose a technique called Pseudo Labeling.

B. Pseudo Labeling

The technique of using a partially trained model to label unlabeled data falls under the category of Pseudo Labeling. Pseudo labeling method uses a small set of labeled data with a large quantity of unlabeled data to enhance the model accuracy [18]. Before bidding for a pseudo labeling process, it is necessary to ensure that the partially trained model performs well during training and validation. Also, the labeled data should be a proper representation of the full data set. Also, there is a possibility of mislabeling the remaining unlabeled data which may cause an adverse effect on the performance of the model. To overcome this issue only the pseudo-labeled samples of a class that obtain a predicted probability that is greater than a particular threshold value is used. Even though this technique does not completely eliminate the risk of mislabeling, it helps to reduce the burden of mislabeled data.

C. Object detection with Transfer Learning

As previously pointed out in Section 1, the most effective technique for the task of object detection is the use of deep convolutional neural networks (CNNs). However, the training time of CNN models on large data sets can be extremely high. A way to overcome this problem is to use weights from pre-trained models which are generated from computer vision benchmark datasets. ImageNet and Microsoft Common Objects in Context (MS COCO) are the most widely used datasets use for computer vision projects[19], [20]. ImageNet contains more than 14 million images with 22000 visual categories and on the other hand, MS COCO provides an accessible object detection image dataset that contains 91 object types with a total of 2.5 million annotated instances in 320,000 images.

Transfer learning is a method that reuses models trained for a similar problem or a slightly different task by fine-tuning the parameters of the pre-trained model[21]. This approach is highly effective for the feature extraction process of CNNs containing several convolution layers followed by max pooling layers[22]. Basically, CNN layers have weight matrices, which are updated during the training process via the backpropagation algorithm [23]. Typically, these multiple forward and backward iterations of the backpropagation algorithm may lead to a high training time. To build the final model, we can directly apply the weights and the model architecture of a pre-trained model trained on large datasets. From a practical perspective, transfer learning can be achieved through (1) training the entire model from scratch or (2) training some layers and leave others frozen. These two approaches depend on the size of the data set and how similar is the new problem to the problem considered for the pre-trained model. A pre-trained model may not be 100% accurate for every application, but it eliminates the huge effort required to build models from scratch.

D. Faster-RCNN

In contrast to object classification models, the object detection and localization models use a bounding box around the object of interest to locate it within the image. Deep learning techniques like Region-based Convolution

Neural Networks (R-CNN) are developed exactly for that purpose, which use selective search to extract regions for an image. Faster R-CNN is an object detection algorithm that is similar to R-CNN. This algorithm utilizes the Region Proposal Network (RPN) that shares full-image convolutional features with the detection network in a cost-effective manner than R-CNN[24]. Instead of using a selective search algorithm as in R-CNN, Faster R-CNN uses a feature map to identify the region proposals and a separate network to predict the region proposals. These region proposals pass through a fully connected convolution layer with softmax classifier to classify the bounding boxes of the image. Fig. 3 illustrates the Faster-RCNN model.

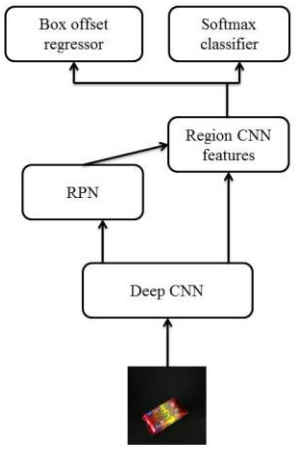

Figure 3 Faster R-CNN model

E. YOLOv3

You Only Look Once or YOLO is a popular algorithm used for object detection and localization [25]. In YOLO a single convolutional network predicts the bounding boxes and the class probabilities for these boxes as shown in Fig. 4. The algorithm divides the image into grids and runs the image classification and localization algorithm on each of the grid cells. It predicts N bounding boxes and confidence scores in each grid. The confidence score reflects the accuracy of the bounding box of that class. Bounding boxes having the class confidence above a threshold value are selected and used to locate the object within the image.

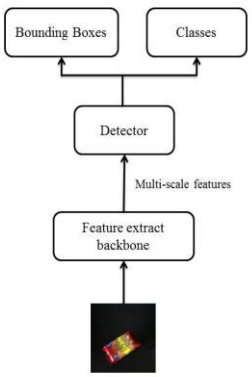

Figure 4 YOLOv3 model

YOLOv3 is an improvement over previous YOLO detection networks [26]. YOLOv3 predicts the coordinates of bounding boxes directly using fully connected layers on top of the convolutional feature extractor. Faster R-CNN object detection model described in Section 2.3 predicts bounding boxes using hand-picked anchor boxes which is somewhat different from YOLOv3 [27]. YOLOv3 uses independent logistic classifiers and binary cross-entropy loss for class prediction.

III. METHOD

In this study, we evaluate the best approach for the detection and localization of overlapping grocery items. This section describes the data collection, the models used, proposed pseudo labeling approach, training models and the evaluation procedure.

A. Data Collection

We captured 5000 images (size of 512x512) of grocery items from 8 different classes, which contain 2000 objects from each class. For the training purposes, we annotated 600 images by using a Visual Object Tagging Tool (VoTT) [28]. Bounding box annotation technique was used as the method to annotate 200 objects from each class. Those classes labeled as “cream cracker”, “sunlight powder”, “milk powder”, “sunlight soap”, “surf excel”, “krisco bites”, “lifebuoy soap”, “signal toothpaste”. The coordinate format of the annotation is defined as (x_{min}, y_{min}, x_{max}, y_{max}). Also, the samples were assigned a label which is a number between 0 and 7. We also create a separate test data set consisting of manually verified samples to reliably evaluate the performance of the models.

B. Models Used

Two models were used in this work to determine which model best suits our task, namely Faster R-CNN with ResNet-101-FPN as backbone architecture (Model 1) and YOLOv3 with a Darknet-53 architecture (Model 2) [29]. We use the PyTorch library to obtain pre-trained models and fine-tunned models [30].

C. Training of the Models

The dataset was split into train sets and validation sets with 90% and 10% of the samples randomly assigned to each set, respectively. We divide the training process into 2 phases. For Training Phase 1, we initialize the models with pretrained weights on the COCO dataset. In that stage, we fixed the backbone architecture with weights and train the classification and regression layers. The size of input images was set to 512x512 and the same hyperparameters were used for the two models. Table 1 shows the hyperparameter values used for the two models.

Table 1 Hyperparameter values used for Model 1 and Model 2

Hyperparameter	Values
Learning rate	4e-4
Training batch size	10
Adam epsilon	le-3
Training epochs	60
Weight decay	0.1

D. Evaluation and Pseudo Labeling

In the field of Data Science, the mean average precision (mAP) is a widely used metric to evaluate the performance of multiclass object detection models [31]. Precision measures how many of the object locations made by the model are actually correct whereas the recall measures how many of the actual object localizations have been predicted by the model. Mean average precision is the average of the area under precision-recall curves (AP) of all classes. Formula 1 shows how mAP is calculated.

$$mAP = \frac{1}{N} \sum_{i=0}^{N} AP_i \qquad (1)$$

We created a manual testing data set to evaluate models. Mean Average Precision (mAP) values of 0.798 and 0.845 were achieved for Faster R-CNN and YOLOv3 models respectively. We observe that both models show satisfactory performance on the testing dataset. Sample detections for non-overlapping grocery items are shown in Fig. 5 in which Fig. 5(a) and Fig. 5(b) are the predictions of YOLOv3 and Faster R-CNN models respectively. Fig. 6 illustrates a couple of sample detections for overlapped and obscured objects. It can be observed that YOLOv3 model (Fig. 6(a)) has performed better in detecting overlapping objects compared to Faster R-CNN model (Fig. 6(b)). Therefore, YOLOv3 model was selected for the pseudo labeling approach (i.e., Training Phase 2).

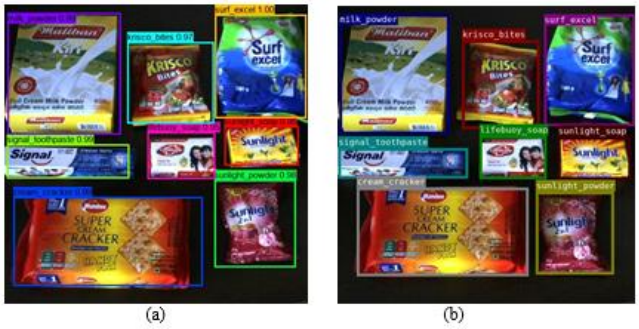

Figure 5 Sample detections for the detection of individual grocery items with non-overlapping boundaries where a) shows the detections of YOLOv3 model and b) shows detections of Faster R-CNN model.

Figure 6 Sample test results for detection of grocery items with overlapping and obscured boundaries where a) shows the detections of YOLOv3 model and b) shows detections of Faster R-CNN model.

.

Fig. 7 illustrates the proposed pseudo labeling process. For this semi-supervised learning method, we used the partially trained YOLOv3 model for pseudo labeling of the remaining 4400 images from the unlabeled data set. In YOLOv3 several convolutional layers are added to the default feature extractor Darknet-53, where the last of these layers predicts the bounding box coordinates, object class and confidence threshold. These 3 output predictions were used for the pseudo labeling approach. Since mislabeling data is known as one of the main drawbacks in the pseudo labeling process, we used a high value of 0.85 as the confidence threshold output. If the model gives the required threshold value, only those object classes and the corresponding bounding box coordinates were used to annotate the images.

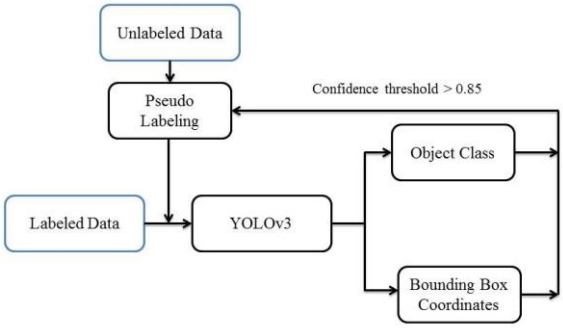

Figure 7 Pseudo labeling process with YOLOv3 model

For Training Process 2 we initialized only the YOLOv3 model and used data augmentation methods to create new training examples[32]. The same hyperparameters with early stopping were applied [33]. Early stopping is a method that allows a model to specify an arbitrarily large number of training epochs and halt training once the model performance stops improving on a hold out validation dataset. We included early stopping to terminate the training if there is no improvement in validation loss after 10 epochs. After Training Process 2 mentioned in Section 3.3, we managed to reach a mAP value of 0.895 for the testing data set. All model evaluations were performed on a single Tesla P40 GPU with use of Pytorch library [33].

IV. Results and Discussion

After Training Process 2 mentioned in Section 3.3, we managed to reach an mAP of 0.895 for the testing data set. Table 2 provides a summary of the mAPs achieved with the models described in Section 3.

Table 2 Testing accuracy for models

Model	mAP
Model 1 (Faster R-CNN with ResNet-101-FPN)	0.798
Model 2 (YOLOv3 with a Darknet-53)	0.845
Model 3 (YOLOv3 with a Darknet-53 + Semi-supervised Pseudo Labeling)	0.895

Comparing the performance of the three models on the test data set, Model 3 (YOLOv3 models with semi-supervised learning) outperforms Model 1 and Model 2. Fig. 8 and Fig. 9 depict the performance of each individual class of Model 2 and Model 3 of each individual class. It can be clearly seen that Model 3 shows better performance for each class compared to Model 2. This is expected since the size of the training data has been increased because of the pseudo labeling process. With the semi-supervised pseudo labeling approach, overall mAP is increased up to 0.895 from 0.845. Also, the average precision of "milk powder", "cream cracker", "sunlight soap", "signal toothpaste" and "sunlight powder" classes are increased. Additionally, there was a significant rise in the average precision (from 0.47 to 0.68) of the "sunlight powder" class. Furthermore, both Model 2 and Model 3 showed a good performance on detecting "surf excel", "milk powder", "cream cracker" and "krisco bites" classes. This is particularly interesting since the primary difference between the two models is the use of pseudo labeling. Since YOLOv3 model is a fast real-time object detector we managed to achieve 30ms of inference time per image detection on Tesla P40 GPU for both of these models. Therefore, the application can be developed for real-world scenarios. However, further work is necessary to increase the average precision of other classes.

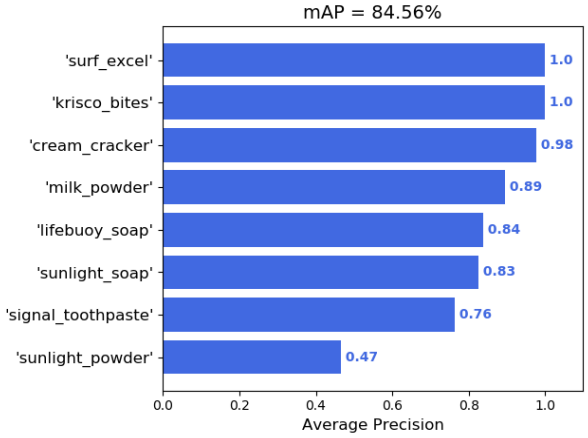

Figure 8 Average precision graph for Model 2 (YOLOv3 with a Darknet-53)

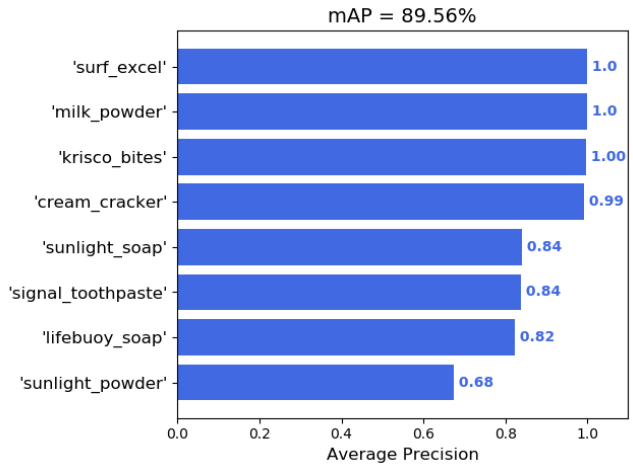

Figure 9 Average precision graph for Model 3 (YOLOv3 with a Darknet-53 + Semi-supervised Pseudo Labeling).

V. Conclusions and Future Work

Recent developments in transfer learning with object detection models have opened various avenues for the application of Computer Vision to real-world tasks. The main contribution of our work is the introduction of a fine-tuned object detection model that can accurately detect multiclass objects with obscure and overlapping boundaries. Additionally, we propose a pseudo labeling technique that can be applied to various domains to extend unlabeled datasets efficiently while minimizing mislabeled samples. We show the best model for overlapping object detection by comparing the performance of YOLOv3 and Faster R-CNN multiclass object detection methods and the effectiveness of semi-supervised learning approach to enhance the accuracy of the model.

This study focuses on bounding box annotations for object detections. Bounding box annotation can be applied to almost any conceivable objects. However, instance and semantic segmentations take object detection a step further. Rather than drawing a bounding box around the objects, instance segmentation annotation goes to pixel-level annotation [34]. Semantic segmentation also assigns pixel-level annotations [35]. Instance segmentation requires the identification and segmentation of individual instances in an image and semantic segmentation requires all the pixels in the image based on their class label. These segmentation methods go further with panoptic segmentation which is a combination of instance and semantic segmentation [36]. In the panoptic segmentation task, we need to classify all the pixels in the image as belonging to a class label, yet also identify what instance of that class they belong to as shown in Fig. 10. Mask R-CNN is an instance segmentation technique that locates each pixel of every object in the image instead of the bounding boxes [37]. We expect to improve the performance of overlapping object detection by using panoptic segmentation with Mask R-CNN technique to increase the object detection accuracy from pixel level for obscure and overlapping boundaries. Future work also includes augmentation of the data set with the pseudo labeling technique to increase the size of the dataset.

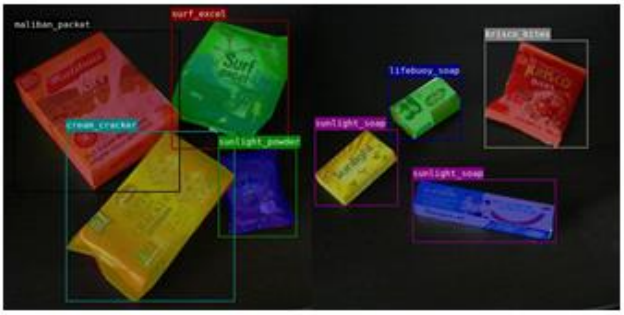

Figure 10 Panoptic segmented data.

REFERENCES

[1] C. Kamusoko, "Image classification," in Springer Geography, 2019.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014.

[3] C. J. Du and Q. Cheng, "Computer vision," in Food Engineering Series, 2014.

[4] M. Shah and R. Kapdi, "Object detection using deep neural networks," in Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems, ICICCS 2017, 2017.

[5] C. Vision and I. P. Toolbox, "Edge and Corner Detection," Computer (Long. Beach. Calif)., 2005.

[6] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," IEEE Transactions on Neural Networks and Learning Systems. 2019.

[7] K. Lenc and A. Vedaldi, "R-CNN minus R," 2015.

[8] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, 2015.

[9] E. Hanna and M. Cardillo, "Faster RCNN," Biol. Conserv., 2013.

[10] D. Yoo, S. Park, J. Y. Lee, A. S. Paek, and I. S. Kweon, "Attentionnet: Aggregating weak directions for accurate object detection," in Proceedings of the IEEE International Conference on Computer Vision, 2015.

[11] W. Liu et al., "SSD: Single shot multibox detector," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016.

[12] 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, 2016.

[13] K. Drid, M. Allaoui, and M. L. Kherfi, "Object detector combination for increasing accuracy and detecting more overlapping objects," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020.

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," Int. J. Comput. Vis., 2010.

[15] K. Sinha, "Semi-supervised learning," in Data Classification: Algorithms and Applications, 2014.

[16] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," Informatica (Ljubljana). 2007.

[17] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019.

[18] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," ICML 2013 Work. Challenges Represent. Learn., 2013.

[19] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., "Imagenet," Adv. Neural Inf. Process. Syst. 25, 2012.

[20] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge," IEEE Trans. Pattern Anal. Mach. Intell., 2017.

[21] W. Transfer, L. Now, T. L. Scenarios, and T. L. Methods, "Transfer Learning - Machine Learning's Next Frontier," PPT, 2017.

[22] H. Nam and B. Han, "Learning Multi-domain Convolutional Neural Networks for Visual Tracking," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

[23] Q. Zhang, Y. N. Wu, and S. C. Zhu, "Interpretable Convolutional Neural Networks," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[24] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High Performance Visual Tracking with Siamese Region Proposal Network," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[25] T. K. Trivedi, M. Glenn, K. Sporer, and G. Hern, "You Only Look Once," Ann. Emerg. Med., 2017.

[26] J. Redmon and A. Farhadi, "Yolov3," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2017.

[27] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in Proceedings of the IEEE International Conference on Computer Vision, 2019.

[28] Microsoft, "VoTT: Visual Object Tagging Tool," GitHub repository, 2018. .

[29] Y. Zhao, R. Han, and Y. Rao, "A new feature pyramid network for object detection," in Proceedings - 2019 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2019, 2019.

[30] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems, 2019.

[31] J. Hui, "mAP (mean Average Precision) for Object Detection," Medium, 2018.

[32] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," J. Big Data, 2019.

[33] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," Constr. Approx., 2007.

[34] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path

Aggregation Network for Instance Segmentation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018.

[35] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang, "Real-Time fusion network for rgb-d sementic segmentation incorporating unexpected obstacle detection," IEEE Robot. Autom. Lett., 2020.

[36] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic segmentation," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., 2020.