Radar Target Recognition Based on Open Set YOLO

G. Meucci[1], S. Ghio, E. Giusti, M. Martorella
National Laboratory of Radar and Surveillance Systems (RaSS) – CNIT
Galleria Gerace 14 56124 Pisa
ITALY

[1](also) University of Pisa, Department of Information Engineering
National Laboratory of Radar and Surveillance Systems (RaSS) – CNIT Pisa
ITALY

giulio.meucci@cnit.it

ABSTRACT

Automatic target recognition (ATR), both for optical and E.M. images, has been a subject of great interest since the last 20 years. The deep learning breakthrough allowed researchers to improve feature extractors by increasing their complexity and since then, traditional classifiers have been outperformed by those based on deep neural network (DNN). So far, DNN-based detectors obtained nearly perfect results on closed sets, namely static datasets, which contain only known classes. Nevertheless, they have a significant decrease in performance when employed in dynamic environment. This problem, often referred to as open set recognition, can be addressed by developing completely new classifiers or by using techniques that exploit a background class. However, few works analyse the possibility of using post-processing methods to adapt a closed set classifier in order to serve as an unknown detector. In this paper, the YOLO model is trained and tested on a dataset of SAR images generated from the MSTAR collection by using targets that are both known and unknown to the network. Two new post-processing methods have been developed making the YOLO detector able to implement the identification of unknown targets.

1.0 INTRODUCTION

During the last decades, radar systems with imaging capabilities have been employed in a wide variety of applications, therefore creating an ever increasing amount of data. Automatic target recognition (ATR) for SAR or ISAR images has consequently gained a lot of interest among researchers. Usually, the ATR problem is divided in two phases, the pre-screening, or definition of the region of interest (ROI), for each target and the classification, which is performed by comparing the ROI with a pre-trained model of the proposed class. Recent works have been using sliding-window detectors with Convolutional Neural Network (CNN) to identify both the ROI and the target class. However, this approach has poor computational efficiency and therefore high inference time. A more efficient solution is represented by the YOLO (You Only Look Once) algorithm [1]. The performance of the YOLO algorithm for known targets have been largely verified on optical images. Due to the success obtained by the YOLO algorithm in the optical domain, recent works started to study its application to radar images [2], [3], [4], thus it is a current and novel area of research in the radar domain. In this paper, the smaller third version of the You Only Look Once (YOLOv3-Tiny) [5] deep neural network, typically trained with optical images, is trained on SAR images from the MSTAR dataset. The YOLO classifier is a fully convolutional neural network (FCNN) and is therefore able to localize and classify multiple targets at once. Furthermore, this network uses independent logistic classifiers to obtain predictions for each class, making it best suited for multilabel datasets. Another problem of ATR is the dynamism of the real world as opposed to the fixed conditions of the training environment. Indeed, most classifiers are developed using the closed set assumption, which implies that the classes used in the training phase are the only ones that the system will encounter during its operational phase. This leads to classifiers that are limited by their training set. On the other hand, open set classification algorithms can label previously unseen targets as unknown and, therefore, better perform when employed in real environments. The identification of an unknown target is often accomplished by using either a confidence threshold on the proposals or a direct comparison between the features of the new target and those of the known classes. The YOLO algorithm has been developed with the closed set assumption. And, to the best of our knowledge, only a few papers tested YOLO in an open set environment using optical images [6] and none tested it on radar ones. Thus, the application of YOLO on an open set radar environment is a novelty of this work. In particular, as demonstrated in our work, this can be done by adding post-processing steps and a hierarchical classification method. In this direction, the developed strategy consists of two methods to detect unknowns from the output of the YOLO closed set classification algorithm. The first newly introduced method exploits the mutually exclusive labels of the dataset and the independent classifiers, introduced in the third version of the YOLO algorithm, to identify as unknown the targets for which the network shows indecision. The second one applies an overlay of new labels to categorize the known targets in the training set instead of classifying them. If correctly defined, the new categories have a better generalization and are therefore more suited to include the unknown targets. Furthermore, since the two layers of labels represent different feature of the targets we can assign different confidence threshold to their predictions, obtaining therefore another degree of freedom in the design of the open set detector. After a brief introduction to the main features of the YOLO architecture we will present the two methods for the identification of unknown targets. The original dataset and the steps taken to obtain the final one will be illustrated in Section 4.1. Section 4.2 shows the metrics used to evaluate the performance and in Section 4.3 the obtained results will be presented and compared.

2.0 YOLO-BASED CLASSIFIERS

YOLO [1], [5], [7] is a one-stage detector for performing object detection in real time on images, videos or live feed. Its network is made of several convolutional layers which extract the features and assign the labels for each target during a single elaboration on the whole image. This is accomplished by dividing the input image in a grid of cells, each cell is then convolved with the filters of the layers to obtain a feature map with reduced spatial resolution but increased semantic meaning. In the tiny version of the algorithm, the one used in the following work, this process is repeated at two different scales to increase the performance of the detector for small sized objects and improve the quality of the information contained in the feature map [5]. Indeed, for each scale the map is concatenated with the output of a previous layer in order to fuse the raw information from the first layers with the more meaningful output of the last ones. Each cell will then evaluate the probability of an object being in it, measured with the objectness score *os*, and the conditional probabilities for all the known classes. The output tensor has the form NxNx((C+4+1)xB) where NxN are the dimensions of the cell grid, C is the number of the known classes and 4+1 are the prediction of the bounding box centre, width and height plus the previously mentioned objectness score, *os*. The last term, B, is the number of bounding box prior assigned to the detectors used at each scale. The bounding box prior are predefined width and height that are computed applying the k-means method to the bounding boxes of the training set. This information are then used by the network as a starting point for the predicted bounding box. The conditional probabilities of belonging to each class are simultaneously predicted through logistic regression and are considered mutually independent. This mechanism was introduced in the third version of the algorithm to improve the network performance in complex dataset with many overlapping labels but, under some assumption, it can help the open set recognition task on simpler domains.

3.0 OPEN SET YOLO – PROPOSED METHODS

3.1 First Method (Indecision Based Classification)

Typically, a closed set classifier in an open set environment will incorrectly assign the unknown targets to known classes with a high confidence and therefore any methods involving a threshold would fail in detecting this event. If we instead assume that the known labels are mutually exclusive, we can repurpose the logistic classifiers of the YOLO model to act as an unknown detector. For example, if our classifier is trained to recognize the models of certain car brands, we are dealing with labels that cannot coexist on the same target and are therefore mutually exclusive. In this scenario, a target labelled as two different models raises the question of whether the network is unable to correctly classify a known class or has been reasonably confused by a target that has never seen before. However, if the classifier performs well on known classes it is unlikely that the network is incorrectly assigning a known label, although still possible. The detected target is therefore probably unknown and can be labelled accordingly.

3.2 Second Method (Hierarchical Classification)

Another approach to the open set classification problem consists in identifying as unknown all the targets that didn't receive a proposal. Even though it may seem the most logical solution to our problem, this idea shifts the complexity of the task from the classification to the detection of the targets. Indeed, since the target is not receiving any proposal, we can't directly rely on the network for the detection of the unknown. A typical solution uses the background class and identifies as unknown all the portions of the input image that have received neither the proposal for the class nor for the background. However, this approach suffers from the high generalization needed for the background class which should ideally include all the non-relevant targets but not the ones that we are trying to identify. Furthermore, as previously stated, closed set classifiers are more likely to assign a known label to the unknown target, especially if it is similar to a known class.

Our approach is closer to the everyday experience. Reusing the previous example, if we see a new model of a car brand, we may not be able to immediately tell the specific name of that car, but we wouldn't label it as "unknown". Instead, we may be able to recognize its brand or at least that it is a car. Organizing the information in a hierarchical structure gives us the ability to categorize new objects without knowing them specifically. Following this idea, we tried to equip the YOLO network with a similar mechanism and added a new layer of labels on top of the original ones in order to create a more abstract representation. The information associated with these new classes are more general and therefore also the generalization of their model should improve, making them more likely to be assigned to an unknown target.

Furthermore, with the introduction of this new layer of generic labels we could require their presence before assigning the specific ones. In our example scenario, if a target is labelled with a specific model name it can also be labelled with a more generic label representing its category, as can be the label "Vehicle". If we require the proposals to contain both these labels we are using a contextual information to enforce the network predictions. The final procedure is represented in Figure 1.

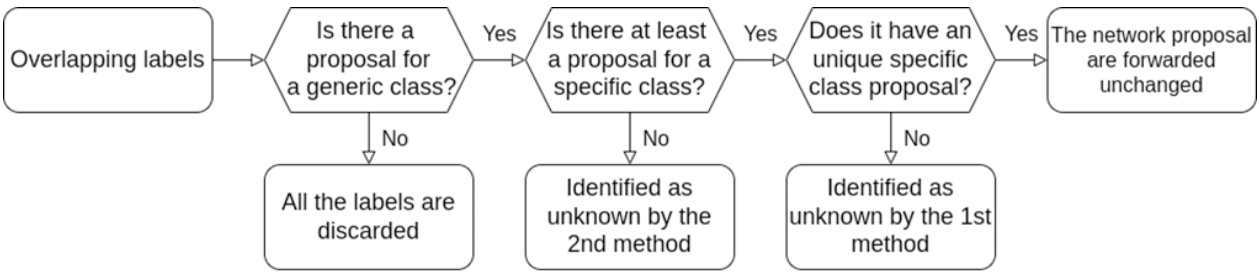

Figure 1: Flowchart of the proposed procedure for the identification unknown targets.

The two presented methods are based on two distinct concepts, one being the absence of specific labels, the other the presence of multiple and mutually exclusive proposals, and therefore can be combined in a single process. Similarly, to the Non-Maximum Suppression (NMS) algorithm we scan the output of the network in search of overlapping predictions and, when found, an analysis on the predicted classes is performed using both the described methods. The final set of unknowns is then the aggregation of the subsets of targets identified by each method. Typically, the minimum confidence threshold for the YOLO predictions is set to 0.25 for all the proposal but, exploiting the overlay of labels introduced for the second method, we can split the threshold and assign a minimum confidence for each layer. Using more than one level of labels led us to a trade off in the design of the open set classifier. The first set of thresholds that we defined requires a lower confidence score (0.1) for the more specific labels, in order to help the indecision-based classification method introduced, and a high confidence score (0.75) for the generic ones. The other set requires high confidence (0.75) for labels that represent highly specific features of the targets, as in the example of the car model mentioned before, and instead accepts less confidence (0.25) for the more generic labels of the new layer. A comparison between these two sets is presented in Section 5.

4.0 EXPERIMENTAL SETUP

4.1 Dataset

The images used to train and test our open set classifier are generated from a subset of the Moving and Stationary Targets Acquisition and Recognition (MSTAR) dataset [8], published by the U.S. Air Force Research Laboratory (AFRL) and Defense Advanced Research Project Agency (DARPA). The SAR images, or chips, collected in the MSTAR represent various military vehicles at 3 different depression angles. We divided the dataset in 3 subsets: the training set which is used to fit the network to the known classes, the validation set which in our case is composed of images of only known classes that the network has never seen and the test set composed only by unknown targets. For the training and validation set we selected only the 2S1, BRDM2, BTR60, D7 and T62 as known classes, since those were the vehicles with more occurrence in the collection. We then divided each vehicles' chips in two groups with proportion 20:1 in order to guarantee the separation between the training and validation set. Those two sets contain only known targets and were used to train the network and to verify the correct training. The test set instead contains only unknowns, namely the ZIL131 and the ZSU234 chips plus the Slicy target's chips, a metal object designed to resemble the other vehicles of the collection and particularly useful to analyse the false alarm rate of the classifier. It is known in the literature [9] that, for SAR images, the portion of the image containing the shadow projected by the target bring as much information as the portion containing the targets itself. For this reason, we included the shadows while creating the bounding boxes and the associated labels as ground truths. For the second method, we added the new label "Vehicle" and assigned it to all the known targets alongside the specific label representing the model of the vehicle (i.e. 2S1, BRDM2, BTR60, D7 and T62 for the known classes and ZIL131 and ZSU234 for the unknown). Despite not being used by the network, we labelled the unknown targets with their specific label to keep track of the proposals for each one. After labelling the targets accordingly to the YOLO format, we resized the chips to 128x128 pixels and randomly applied data augmentation techniques such as flips, circular shift and reflections. We also added gaussian noise to the training set and compared the performance of the detector while varying the standard deviation. The final images of the dataset are generated as a 3x3 composition of MSTAR images and two-dimensional realization of noise with standard deviation equal to the original background of the surrounding chips. We applied this procedure in order to obtain a more realistic dataset with scenes containing a random number of targets in various positions instead of having them always at the centre of the image, as in the original MSTAR chips. Figure 2 is an example of the generated images.

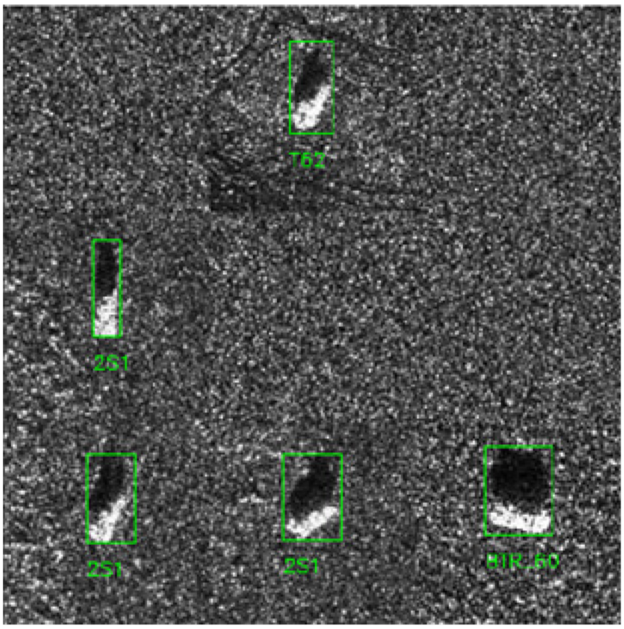

Figure 2: Example image taken from the training set showing 3 x 3 composition generated using the MSTAR chips and the bidimensional noise realizations.

4.2 Performance Metrics

We measured the accuracy of the detector with the Intersection over Union (IoU) [10] metric, which is often used in object detection challenges since any algorithm that predicts bounding boxes can be evaluated using such metric. We can obtain the IoU score comparing the ground truth box and the network's prediction with:

$$IoU = \frac{Intersection\ area}{Union\ area}$$

Another widely used metrics in object detection is the mean Average Precision (mAP) [11] and is defined as the average over all the classes of the Average Precision (AP). In turn, the AP is the area under the precision-recall curve, where precision and recall are respectively defined as:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Where we defined as TP the True Positive, as FP the False Positive and as FN the False Negative. The mAP is particularly suited to measure the performance of a classification algorithm since it can be combined with the IoU to obtain a single value which accounts for the accuracy of both the detector and the classifier.

5.0 RESULTS

5.1 Description of the Training Parameters

In this Section after a description of the training parameters, the obtained results will be presented. A common problem of training deep neural network on small datasets is overfitting. To address this, we used transfer learning and started the re-training of the whole network from the weights of the YOLOv3 trained on the COCO dataset, available on the author site [12]. With few preliminary experiments, involving only the known classes, we tuned the hyperparameters of the network in order to maximize the mAP score. Following [13] we also reduced the batch size from 64 to 16 and set the batch subdivision to 4 in order to improve the model's generalization and reduce the computational requirements for the training phase. We used the step decay schedule with factor 1/10 to update the learning rate at iterations 50000 and 80000. The latter didn't trigger any change in the loss function, so we stopped the training at iteration 100000.

5.2 Results Without Noise Injection

In the following subsection we present the results obtained on the dataset generated without adding Gaussian noise in the training set for both the confidence thresholds set mentioned in Section 3. The first set requires a minimum confidence score equal to 0.75 for the "Vehicle" class and 0.1 for the other known classes. The low confidence for the specific labels of the known classes increases the probability of having more than one label for each target and therefore increases the probability of detecting the unknown with the indecision-based method. The second set instead requires 0.25 for the "Vehicle" class and 0.75 for the known ones. This setup better support the second method since the high confidence required for the specific labels filters out some of the predictions, thus leaving the targets with the "Vehicle" label only.

The results Obtained on the validation set for the two sets of thresholds are reported in Table 1 and in Table 2. The minimum required IoU score is set to 0.75.

Table 1: Validation set – first set of thresholds.

	2S1	**BRDM2**	**BTR60**	**D7**	**T62**
Labeled correctly	207	182	154	217	159
AP Score	0.86	0.87	0.86	0.85	0.77
Total			919		
Labeled unknown	0	1	0	0	0
by 1st method	0	0	0	0	0
by 2nd method	0	1	0	0	0
Total			1		
Miss	33	44	22	37	47

Table 2: Validation set – second set of thresholds.

	2S1	**BRDM2**	**BTR60**	**D7**	**T62**
Labeled correctly	207	182	154	217	159
AP Score	0.86	0.87	0.86	0.85	0.77
Total			919		
Labeled unknown	1	17	0	2	2
by 1st Method	0	0	0	0	0
by 2nd Method	1	17	0	2	2
Total			22		
Miss	32	28	22	35	45

Table 3 and Table 4 report instead the results obtained on the test set, containing the unknowns and the Slicy target, for the respective set of thresholds. The number of false alarms raised by the fake target (i.e. the Slicy) are not included in the total count of known and unknown declarations but are instead reported aside between brackets. Since this dataset contains only unknown targets whose bounding box prior are not known to the network, we lowered the IoU minimum score to 0.5.

Table 3: Test set – first set of thresholds.

	ZSU234	ZIL131	Slicy
Wrongly labeled as known	217	190	
Total:	407		(15)
Correctly labeled unknown	41	58	
by 1st Method	41	58	5
by 2nd Method	0	0	0
Total:	99		(5)
Miss	11	15	91

Table 4: Test set – second set of thresholds.

	ZSU234	ZIL131	Slicy
Wrongly labeled known	209	230	
Total	439		(10)
Correctly labeled unknown	56	18	
by 1st Method	0	1	0
by 2nd Method	56	17	49
Total	74		49
Miss	4	15	52

However, with a low confidence for the specific labels, as in the case of the first set of thresholds, the number of false alarms raised by the class Slicy would increases. To address this issue, we required the label “Vehicle” to be assigned along the specific one. Since the generic label “Vehicle” requires a high confidence to be assigned, we widely reduced the overall number of false targets detected without decreasing the performance on the other classes.

5.3 Results with Noise Injection

For the noise analysis we set the standard deviation to 5 and 10 but, to facilitate comparison, also the results obtained without noise addition will be reported. The corresponding SNR is evaluated on the whole dataset. At every increase of standard deviation, the dataset is generated again and the whole network is retrained. The test set is left unchanged to facilitate the comparison of the proposed methods. Table 5 and Table 6 report, for the validation set and for each thresholds set described in the previous subsection: the overall number of targets, the mAP, the number of targets wrongly identified as unknown by each method and the number of missed targets (Miss).

Instead, for the test set, Table 7 and Table 8 report: the number of unknown wrongly assigned to a known class (Known), the number of false alarm raised by the Slicy target (FA), the total number of unknown correctly identified by each method and the number of missed targets (Miss).

Table 5: Validation set – first set of thresholds.

STD (SNR dB)	0 (16.40)	5 (16.33)	10 (16.00)
N° targets	1 103	1 100	1 090
mAP	0.83	0.86	0.83
1st Method	2	4	2
2nd Method	0	0	0
Miss	185	153	187

Table 6: Validation set – second set of thresholds.

STD (SNR dB)	0 (16.40)	5 (16.33)	10 (16.00)
N° targets	1 103	1 100	1 090
mAP	0.82	0.85	0.81
1st Method	0	0	0
2nd Method	22	12	16
Miss	162	143	172

Table 7: Test set – first set of thresholds.

STD (SNR dB)	0 (16.40)	5 (16.33)	10 (16.00)
Known	407	368	354
FA	20	14	35
1st Method	99	119	141
2nd Method	0	0	1
Miss	137	156	147

Table 8: Test set – second set of thresholds.

STD (SNR dB)	0 (16.40)	5 (16.33)	10 (16.00)
Known	439	310	253
FA	59	68	85
1st Method	1	0	0
2nd Method	73	202	276
Miss	130	131	114

6.0 CONCLUSIONS

The aim of this preliminary study was to identify new methods for adapting a closed set algorithm, the YOLO network, to act as an open set detector, able to recognize targets never seen before. Since the MSTAR dataset contains SAR images of similar military vehicles, we couldn't directly rely on the network for the identification of unknown targets that, in most cases, were assigned to known classes. We then defined two post-processing methods that analyse the output of the YOLO classifier in search of contradictory labels or missing information. Despite being developed having in mind the MSTAR dataset, the proposed methods can be adapted to work in any dataset. Indeed, the hierarchical classification is nothing more than a simplified representation of our way of organizing information. The indecision method instead is based on a stronger assumption which involves working with mutually exclusive labels. In the majority of applications this is not true but it could still be possible to identify a subset of known classes that cannot coexist together and we could then modify the method in order to work only on that particular subset. In Table 2, where we used the second set of thresholds of 0.75 and 0.25 for the "Vehicle" and the known classes respectively, we can notice a significant increase in the number of known target wrongly identified as unknown. However, only the number of missed targets decreased and therefore the introduced method improved the detection accuracy also for the known classes. This is due to the greater generalization obtained for the "Vehicle" class which, although less informative, has successfully located more targets than the specific classes. This behaviour can be helpful in scenarios where the system needs to detect every target as for autonomous driving or for military applications. The experiments performed on the test set show that both methods successfully identified a portion of the unknowns. However, the vast majority of the unknowns were misclassified as known targets and for the second set of thresholds of Table 4 a significant number of the Slicy targets are assigned to the "Vehicle" class. The addition of Gaussian noise to the training set slightly influenced the performance of the classifier on the validation set as showed in Table 5 and Table 6. Nevertheless, the results obtained on the test set significantly improved with respect to the ones showed in Table 3 and Table 4. Indeed, the number of unknown targets correctly identified increases with the standard deviation of the injected noise but, for the first set of thresholds, also the number of missed targets increases or, for the second set of thresholds, the number of false alarms increases. In conclusion, the first presented method along with the associated confidence thresholds proved its reliability in terms of false alarm rate but the identification of most of the unknowns failed. The hierarchical classification with the opposite set of thresholds showed instead a high false alarm rate, assigning the Slicy target to the "Vehicle" class 3 times out of 4. However, for STD equal to 10 in Table 8, 40% of unknown targets are correctly identified. This results can be used as a starting point for the development of an adaptive classifier able to recognize the limits of its knowledge and learn.

7.0 REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi (2016). "You Only Look Once: Unified, Real-Time Object Detection".

[2] G. Li, Z. Song and Q. Fu, "Small Boat Detection for Radar Image Datasets with YOLO V3 Network," 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019, pp. 1-5.

[3] Kim, W.; Cho, H.; Kim, J.; Kim, B.; Lee, S. YOLO-Based Simultaneous Target Detection and Classification in Automotive FMCW Radar Systems. Sensors 2020, 20, 2897.

[4] M. Woollard, M. Ritchie and H. Griffiths, "Investigating the effects of bistatic SAR phenomenology on feature extraction," 2020 IEEE International Radar Conference (RADAR), 2020, pp. 906-911.

[5] J. Redmon and A. Farhadi (2018). "YOLOv3: An Incremental Improvement".

[6] R. Dhamija, M. Günther, J. Ventura and T. E. Boult (2020). "The Overlooked Elephant of Object Detection: Open Set," 2020 IEEE Winter Conference on Applications of Computer Vision (WACV) pp. 1010-1019.

[7] J. Redmon and A. Farhadi (2016). "YOLO9000: Better, Faster, Stronger".

[8] AFRL and DARPA, "Sensor data management system website, MSTAR database", 1995. [Online]. Available: https://www.sdms.afrl.af.mil/index.php?collection=mstar. [Accessed: 23-Sept-2021].

[9] S. Papson and R. M. Narayanan, "Classification via the Shadow Region in SAR Imagery," in IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 2, pp. 969-980.

[10] A. Rosebrock, "Intersection over Union (IoU) for object detection", (2016). [Online]. Available: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection. [Accessed: 23-Sept-2021]

[11] J. Hui, "mAP (mean Average Precision) for Object Detection", (2018). [Online]. Available: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173. [Accessed 23-Sept-2021]

[12] J. Redmon, "YOLO: Real-Time Object Detection", (2018). [Online]. Available: https://pjreddie.com/darknet/yolo. [Accessed 23-Sept-2021]

[13] D. Masters, C. Luschi (2018). "Revisiting small batch training for deep neural networks".