Toward an ensemble of object detectors.

DANG, T., NGUYEN, T.T. and MCCALL, J.

2020

Toward an Ensemble of Object Detectors

Truong Dang[1], Tien Thanh Nguyen[1], and John McCall[1] ✉

School of Computing, Robert Gordon University, Aberdeen, UK

Abstract. The field of object detection has witnessed great strides in recent years. With the wave of deep neural networks (DNN), many breakthroughs have achieved for the problems of object detection which previously were thought to be difficult. However, there exists a limitation with DNN-based approaches as some architectures are only suitable for particular types of object. Thus it would be desirable to combine the strengths of different methods to handle objects in different contexts. In this study, we propose an ensemble of object detectors in which individual detectors are adaptively combine for the collaborated decision. The combination is conducted on the outputs of detectors including the predicted label and location for each object. We proposed a detector selection method to select the suitable detectors and a weighted-based combining method to combine the predicted locations of selected detectors. The parameters of these methods are optimized by using Particle Swarm Optimization in order to maximize mean Average Precision (mAP) metric. Experiments conducted on VOC2007 dataset with six object detectors show that our ensemble method is better than each single detector.

Keywords: Object detection · Ensemble method · Ensemble learning · Evolutionary Computation · Particle Swarm Optimization

1 Introduction

Object detection is a problem in which a learning machine has to locate the presence of objects with a bounding box and types or classes of the located objects in an image. Before the rise of Deep Neural Networks (DNN), traditional machine learning methods using handcrafted features [13,22] were used with only modest success since these extracted features are not representative enough to describe many kinds of diverse objects and backgrounds. With the successes of DNN in image classification [11], researchers began to incorporate insights gained from Convolutional Neural Networks (CNN) to object detection. Some notable results in this direction include Faster RCNN [7] or You Look Only Once (YOLO) [16]. However, some object detectors are only suitable for specific types of objects. For example, YOLO struggles with small objects due to strong spatial constraints imposed on bounding box predictions [15]. In this study, we propose to combine several object detectors into an ensemble system. By combining multiple learners for the collaborated decision, we can obtain better results than using a single learner [20]. The key challenge of building ensembles of object

detectors is to handle multiple outputs so that the final output can determine what objects are in a given image and where they are located.

The paper is organized as follows. In section 2, we briefly review the existing approaches relating to object detection and ensemble learning. In section 3, we propose a novel weight-based ensemble method to combine the bounding box predictions of selected base detectors. The bounding boxes for combination are found by a greedy process in which boxes having Intersection-over-Union (IoU) values with each other higher than a predetermined threshold are grouped together. We consider an optimisation problem in maximizing the mean Average Precision (mAP) metric of the detection task. The parameters of combining method are found by using an evolutionary computation-based algorithm in solving this optimisation problem. The details of experimental studies on the VOC2007 dataset [6] are described in section 4. Finally, the conclusion is given in section 5.

2 Background and Related Work

2.1 Object Detectors

Most early object detection systems were based on extracting handcrafted features from given images then applying a a conventional learning algorithm such as Support Vector Machines (SVM) or Decision Trees [13, 22] on those features. The most notable handcrafted methods were the Viola-Jones detector [21] and Histogram of Oriented Gradients (HOG) [5]. However, these methods only managed to achieve modest accuracy while requiring great expertise in handcrafting feature extraction. With the rise of deep learning, in 2014 Girshick et al. proposed Regions based on Convolutional Neural Network (CNN) features (called RCNN), the first DNN-based approach for object detection problem [8]. This architecture extracts a number of object proposals by using a selective search method and then each proposal is fed to a CNN to extract relevant features before being classified by a linear SVM classifier. Since then, object detection methods have developed rapidly and fall into two groups: two-stage detection and one-stage detection. Two-stage detection such as Fast-RCNN [7] and Faster-RCNN [17] follows the traditional object detection pipeline, generating region proposals first and then classifying each proposal into each of different object categories. Even though these networks give promising results, they still struggle with objects which have a broad range of scales, less prototypical images, and that require more precise localization. One-stage detection algorithms such as YOLO [15] and SSD [12] regard object detection as a regression or classification problem and adopt a unified architecture for both bounding box localization and classification.

2.2 Ensemble methods and optimization

Ensemble methods refer to the learning model that combines multiple learners to make a collaborated decision [18, 20]. The main premise of ensemble learning

is that by combining multiple models, the prediction of a single learner will likely be compensated by those of others, thus making better overall predictive performance. Nowadays, many ensemble methods have been introduced and they are categorized into two main groups, namely homogeneous ensembles and heterogeneous ensembles [20]. The first group includes ensembles generated by training one learning algorithm on many schemes of the original training set. The second group includes ensembles generated by training several different learning algorithms on the original training set.

Research on ensemble methods focuses on two stages of building an ensemble, namely generation and integration. For the generation stage, approaches focus on designing novel architectures for the ensemble system. Nguyen et al. [19] designed a deep ensemble method that involves multiple layers of ensemble of classifiers (EoC). A feature selection method works on the output of a layer to obtain the selected features as the input for the next layer. In the integration stage, besides several simple combining algorithms like Sum Rule and Majority Vote [10], Nguyen et al. [20] represented the predictions of the classifiers in the form of vectors of intervals called granule prototypes by using information granules. The combining algorithm then measures the distance between the predictions for a test sample and the granule prototypes to obtain the predicted label. Optimization methods have been applied to improve the performance of existing ensemble systems in terms of ensemble selection (ES) which aims to search for a suitable EoC that performs better than using the whole ensemble. Chen et al. [2] used ACO to find the optimal EoC and the optimal combining algorithm.

3 Proposed Method

3.1 General Description

In this study, we introduce a novel ensemble of object detectors to obtain higher performance than using single detectors. Assume that we have T base object detectors, denoted by $OD_i (i = 1, ..., T)$. Each detector works on an image to identify the location and class label of objects in the form of prediction results $\mathbf{R_i} = \{R_{i,j}\}, R_{i,j} = \Big(BB_{i,j}, (l_{i,j}, conf_{i,j})\Big), (i = 1, ..., T; j = 1, ..., r_i$ where r_i is the number of objects detected by $OD_i)$. The elements of $R_{i,j}$ are detailed as:

- Bounding box $BB_{i,j} = \big(x_{i,j}, y_{i,j}, w_{i,j}, h_{i,j}\big)$ identifies the location of a detected object where $x_{i,j}, y_{i,j}, w_{i,j}$ and $h_{i,j}$ are the top-coordinates and the width and height of the bounding box
- Prediction $\big(l_{i,j}, conf_{i,j}\big)$ where $l_{i,j}$ is the predicted label and $conf_{i,j}$ is the confidence value, which is defined as the probability for the prediction of this label

Our proposed ensemble algorithm deals with the selection of suitable detectors among all given ones, as well as combining the bounding boxes of the

selected detectors. In order to select suitable detectors, we introduce a number of selection variables $\alpha_j \in \{0,1\}, j = 1, ..., T$ with each binary variable α_j representing whether detector OD_j is selected or not. The combining process is conducted after the selection process. To combine the bounding boxes made by the selected detectors, we need to know which bounding box of each detector predicts the same object. Our proposed method consists of two steps:

- Step 1: Measure the similarity between pairs of bounding boxes between the detection results from different detectors to create groups of similar bounding boxes
- Step 2: For each group, combine the bounding boxes

The similarity between bounding boxes is measured using Intersection over Union (IoU), which is very popular in object detection research [22]. With two bounding boxes $BB_{i,j}$ and $BB_{p,q}$, the IoU measure between them is given by:

$$IoU(BB_{i,j}, BB_{p,q}) = \frac{area(BB_{i,j} \cap BB_{p,q})}{area(BB_{i,j} \cup BB_{p,q})} \tag{1}$$

This measure is compared to a threshold θ ($0 \leq \theta \leq 1$). If the $IoU > \theta$ then they are grouped together, eventually forming a number of box groups $G = (g_1, g_2, ..., g_K)$, where K is the number of groups. Note that we do not consider the $IoUs$ between boxes made by the same detector ($i \neq p$) since we combine bounding boxes of different detectors. We also combine bounding boxes that have the same predicted label. For each group, we perform combination of the bounding boxes. Let $W_i^x, W_i^y, W_i^w, W_i^h \in [0,1]$ be the weights of detector $OD_i (i = 1, ..., T)$. Then the combined bounding box for group g_k will be $BB_k = (x_k, y_k, w_k, h_k)$ in which:

$$coord_k = \frac{\sum_{BB_{p_l(k),q_l(k)} \in g_k} \mathbb{I}[\alpha_{p_l(k)} = 1] W_{p_l(k)}^{coord} coord_{p_l(k),q_l(k)}}{\sum_{BB_{p_m(k),q_m(k)} \in g_k} W_{p_m(k)}^{coord}}, \tag{2}$$

where $\mathbb{I}[.]$ is the indicator function, and $coord_k \in \{x_k, y_k, w_k, h_k\}$. Therefore, our ensemble is completely determined by the following parameters: $(W_i^x, W_i^y, W_i^w, W_i^h, \alpha_j, \theta), i, j = 1, ..., T$

3.2 Optimisation

The question that arises from the proposed method is how to search for the best parameters $(W_i^x, W_i^y, W_i^w, W_i^h, \alpha_j, \theta), i, j = 1, ..., T$ for each situation, where $W_i^x, W_i^y, W_i^w, W_i^h$ are the bounding box weights, α_j are the selection variables, and θ is the IoU threshold. We formulate an optimisation problem which we can solve to find the optimal value for these parameters. The fitness function is chosen to be the mean Average Precision (mAP), which is defined as the average of Average Precision for each class. In order to calculate AP_c, we need to calculate the precision and recall. Precision and recall are defined as follows:

Algorithm 1 Combining object detectors

Input: Bounding box results by the detectors (BB_i), the prediction labels (l_i), confidence values ($conf_i$), index of detector (det_i) ($i = 1, ..., nbb$) with nbb being the total number of bounding boxes, bounding box weights for each detector ($W_j^x, W_j^y, W_j^w, W_j^h$), the threshold for choosing each detector α_j and IoU threshold θ

Output: The combined bounding boxes

Remove detectors that does not satisfy $\alpha_j \geq 0.5$. Sort the bounding boxes in descending order of confidence value. Set $G \leftarrow \{\}, E \leftarrow \{\}, Assign \leftarrow \{assign_1, assign_2, ..., assign_{nbb}\}$ where $assign_i$ is the group which BB_i is assigned to, and initialize $assign_i$ to 0. Set $group\_idx \leftarrow 1$.
for $i \leftarrow 1$ to nbb **do**
  if $assign_i \neq 0$ **then**
    continue
  $assign_i \leftarrow group\_idx$
  for $j \leftarrow i + 1$ to nbb **do**
    if $assign_j \neq 0$ or $det_i == det_j$ or $l_i \neq l_j$ **then**
      continue
    if $IoU(BB_i, BB_j) > \theta$ **then**
      $assign_j \leftarrow group\_idx$
  $group\_idx \leftarrow group\_idx + 1$
$K \leftarrow groupd\_idx - 1$
$G \leftarrow \{g_1, g_2, ..., g_K\}$ where $g_k = \{BB_i\}$ such that $assign_i == k$
for $k \leftarrow 1$ to K **do**
  Combine boxes in g_k to get $BB_k = (x_k, y_k, w_k, h_k)$ by using Eq. 2,
  $E.insert(BB_k)$
return E

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN} \tag{3}$$

where TP (True Positive) is the number of correct cases, FP (False Positive) is the number of cases where a predicted object does not exist, FN (False Negative) is the number of cases where an object is not predicted. The IoU measure between a predicted bounding box and a ground truth box determines whether the ground truth box is predicted by the algorithm. The AP summarises the shape of the precision/recall curve, and is evaluated by firstly computing a version of the measured precision/recall curve with precision monotonically decreasing, by setting the precision for recall r to the maximum precision obtained for any recall $r^{'} \geq r$. Then the AP is calculated as the area under this curve by numerical integration. This is done by sampling at all unique recall value at which the maximum precision drops. Let p_{interp} be the interpolated precision values. Then the average precision is calculated as follows:

$$AP = \sum_n (r_{n+1} - r_n) p_{interp}(r_{n+1}), p_{interp}(r_{n+1}) = max_{r_1 \geq r_{n+1}}(p_1) \tag{4}$$

Thus with T detectors, the optimisation problem is given by:

$$\begin{aligned} &max_{\mathbf{w}=(W_i^x,W_i^y,W_i^w,W_i^h,\alpha_j,\theta)} mAP(\mathbf{w}) \\ &\text{s.t. } W_i^x, W_i^y, W_i^w, W_i^h \in [0,1], \alpha_j \in \{0,1\}, \theta \in [0,1], i,j = 1,\cdots,T \end{aligned} \tag{5}$$

We use PSO [3, 9] to find the optimal values for $(W_i^x, W_i^y, W_i^w, W_i^h, \alpha_j, \theta)$. Compared to other optimisation algorithms, PSO offers some advantages. Firstly, as a member of the family of evolutionary computation methods, it is well suited to handle non-linear, non-convex spaces with non-differentiable, discontinuous objective functions. Secondly, PSO is a highly-efficient solver of continuous optimisation problems in a range of applications, typically requiring low numbers of function evaluations in comparison to other approaches while still maintaining quality of results [14]. Finally, PSO can be efficiently parallelized to reduce computational cost. To work with continuous variables in PSO, we convert each α_j into a continuous variable belonging to $[0, 1]$. If α_j is higher than 0.5, the corresponding detector is added to the ensemble. The average mAP value in a 5-fold cross-validation procedure is used as the fitness value.

The combining and training procedures are described in Algorithm 1. Algorithm 1 receives inputs including the bounding boxes made by the detectors (BB_i), confidence values ($conf_i$), prediction labels (l_i) and the parameters $(W_i^x, W_i^y, W_i^w, W_i^h, \alpha_j, \theta)$. Each bounding box ($BB_i$) also has an associated variable (det_i) which delineates the index of the detector responsible for (BB_i). For example, if (BB_i) is predicted by the detector (OD_j) then $det_i = j$. Line 1 sorts the selected bounding boxes in decreasing order of confidence value. Line 3-10 assigns each bounding box to a group. For each bounding box BB_i we first check if it has been assigned to one of the existing groups before assigning it to the new group *group_idx* (line 3-5). Then with each unassigned bounding box BB_j that is not made by the same detector as that of BB_i and have the same prediction we add BB_j to group *group_idx* if its IoU value with BB_i is greater than θ (line 6-10). After all boxes are grouped, lines 12 to 17 combine the boxes in each group and returns the combined bounding boxes.

4 Experimental Studies

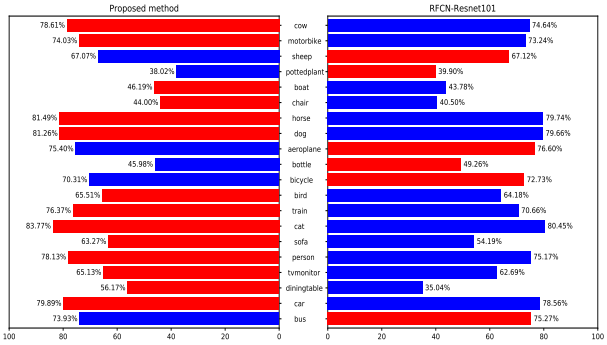
4.1 Experimental Setup

In the experiments, we used a number of popular object detection algorithms as base detectors for our ensemble method. The base detectors used are SSD Resnet50, SSD InceptionV2, SSD MobilenetV1 [12], FRCNN InceptionV2, FRCNN Resnet50 [17], and RFCN Resnet101 [4]. We used the default configuration for all of these methods. Training process was done for 50000 iterations. For the PSO algorithm, the inertial weight a was set to 0.9 while two parameters C_1 and C_2 were set to 1.494. The number of iterations was set to 100 while the population size was set to 50. The dataset VOC2007 was used in this paper containing 5011 images for training and validation, and 4952 images for testing. The evaluation metric used in the paper was mAP (mean Average Precision). Among the 9963 images in the VOC2007 dataset, there are 2715 images having at least one

Table 1. Left: mAP result for the base detectors and the proposed method. Right: Weights for the bounding boxes of each base detectors (x, y, w, h, α)

Detector	mAP (%)
RFCN-Resnet101	64.67
FRCNN-InceptionV2	62.02
SSD-InceptionV2	41.96
SSD-Mobilenet-V1	38.4
SSD-Resnet50	39.93
FRCNN-Resnet50	64.34
Proposed method	**67.23**

Detector \ Weights	x	y	w	h	α
RFCN-Resnet101	0.77	0.56	0.32	0.33	0.76
FRCNN-InceptionV2	0.77	1.00	1.00	0.71	0.53
SSD-InceptionV2	0.49	0.71	0.22	0.30	0.93
SSD-Mobilenet-V1	0.35	0.93	0.25	0.00	0.42
SSD-Resnet50	0.00	0.00	0.94	0.58	0.73
FRCNN-Resnet50	0.89	0.80	0.29	1.00	0.99



The red or blue color means better or poorer performance on an object

Fig. 1. A comparison of AP result for each class between the proposed method and RFCN-Resnet101

object of difficult tag. Because we focus on the improvements of combining the results of bounding boxes from each detector, the difficult examples have been included into the evaluation.

4.2 Result and discussion

Table 1 (left) shows the mAP result of the proposed method and the base detectors. The proposed method has mAP value of 67.23%, which outperforms the best base detector RFCN-Resnet101 by 2.56%. Figure 1 shows a detailed comparison of AP values between the two methods for each class. It can be seen that the proposed method achieves a remarkable increase for the "dining table" object, from 35.04% to 56.17%. This is followed by "sofa" with an increase of 9.08% from 54.19% to 63.27%. Other objects such as "dog" or "train" also saw a modest increase. On the other hand, "bicycle" and "bottle" saw a decrease, from 72.73% to 70.31% and from 49.26% to 45.98% respectively. It should be noted that ensemble methods ensure that the overall result is better, even though some cases might be worse than the base learners. In total, there are 14 object types

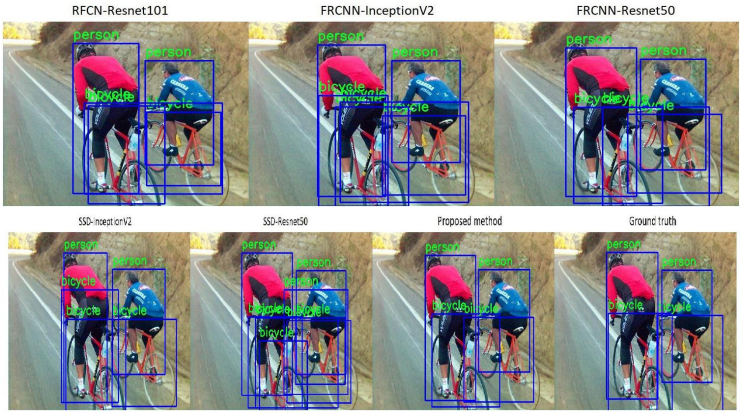


Fig. 2. Example result of selected base detectors and proposed method

that saw an increase due to the proposed method. Table 1 (right) shows the weights for the bounding boxes for each of the base detectors. The first and second columns show the weights of the top-left coordinates, while the third, fourth and fifth columns show the width, height weights and selection threshold respectively. From the table, it is clear that the algorithm automatically chooses the better base detectors for combining the bounding boxes, since most of the contributions of weights are from RFCN-Resnet10 (64.67%), FRCNN-InceptionV2 (62.02%) and FRCNN-Resnet50 (64.34%).

Figure 2 provides a comparison between the selected base detectors (those with $\alpha_i \geq 0.5$ after optimisation) and the proposed method. It can be seen that RFCN-Resnet101, SSD-Resnet50, and FRCNN-Resnet50 correctly identify two bicycles, but wrongly predicts another bicycle that spans the two real bicycles. On the other hand, FRCNN-Resnet50 wrongly predicts three person objects in the image. Due to the combination procedure, the redundant bicycle and person objects have been removed. Also, the bounding box for the left person by SSD-InceptionV2 is slightly skewed to the right, but after applying weighted sum of bounding boxes of the base detectors, the combined box has been positioned more accurately.

5 Conclusion

In this paper, we presented a novel method for combining a number of base object detectors into an ensemble that achieves better results. The combining method is constructed using PSO algorithm to search for a defining parameter set that optimise mAP. Parameters are selective indicators which show whether detectors are selected or not. The bounding boxes of selected detectors then are combined based on a weights-based combining method. Our results on a benchmark dataset show that the proposed ensemble method is able to combine

the strengths and mitigate the drawbacks of the base detectors, resulting in an improvement compared to each individual detector.

References

1. Banfield, R., Hall, L., Bowyer, K., Kegelmeyer, W.: Ensemble diversity measures and their application to thinning. Information Fusion **6**, 49–62 (03 2005)
2. Chen, Y., et al.: Applying ant colony optimization to configuring stacking ensembles for data mining. Expert Syst. Appl. **41**(6), 2688–2702 (May 2014)
3. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. Trans. Evol. Comp **6**(1), 58–73 (Feb 2002)
4. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Proceedings of NIPS. p. 379–387 (2016)
5. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Proceedings of CVPR. vol. 1, pp. 886–893 (2005)
6. Everingham, M., et al.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007), `http://host.robots.ox.ac.uk/pascal/VOC/`
7. Girshick, R.: Fast R-CNN. In: Proceedings of ICCV. pp. 1440–1448 (2015)
8. Girshick, R., Donahue, J., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of CVPR. pp. 580–587 (2014)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IJCNN. vol. 4, pp. 1942–1948 (Nov 1995)
10. Kittler, J., Hatef, M., et al.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(3), 226–239 (Mar 1998)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Proceedings of NIPS, Curran Associates, pp. 1097–1105 (2012)
12. Liu, W., Anguelov, D., et al.: SSD: Single Shot MultiBox Detector. In: Proceedings of ECCV. pp. 21–37. Springer (2016)
13. Lowe: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **60**(2), 91–110 (Nov 2004)
14. Perez, R., Behdinan, K.: Particle swarm approach for structural design optimization. Computers & Structures **85**, 1579–1588 (10 2007)
15. Redmon, J., et al.: You only look once: Unified, real-time object detection. In: Proceedings of CVPR. pp. 779–788 (2016)
16. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. arXiv:1804.02767 [cs] (Apr 2018)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (06 2015)
18. T. Nguyen, M. Nguyen, C. Pham, C. Liew, P. Witold: Combining heterogeneous classifiers via granular prototypes. Applied Soft Computing **73** (09 2018)
19. T. Nguyen, T. Dang, T. Pham, L. Dao, V. Luong, J. McCall, C. Liew: Deep heterogeneous ensemble. In: Proceedings of ICONIP. pp. 1–9 (2019)
20. T. Nguyen, V. Luong, T. Dang, C. Liew, J. McCall: Ensemble selection based on classifier prediction confidence. Pattern Recognition **100**, 107104 (2020)
21. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of CVPR. vol. 1, pp. 511–518 (2001)
22. Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X.: Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems pp. 1–21 (2019)