YOLOv3 Precision Improvement by the Weighted Centers of Confidence Selection

Adrian Horzyk
Department of Biocybernetics and Biomedical Engineering
AGH University of Science and Technology in Krakow
Krakow, Poland
horzyk@agh.edu.pl, ORCID: 0000-0001-9001-4198

Efe Ergün
Department of Computer Science
AGH University of Science and Technology in Krakow
Krakow, Poland
efeergun96@gmail.com, ORCID: 0000-0002-9653-2057

Abstract—**One of the most popular and widely used object detection algorithm today is the YOLOv3 due to its high performance and speed. However, YOLOv3 is not the best algorithm in terms of precision. This paper introduces a substantial change to the post-processing routine of the YOLOv3 after the prediction to increase its final accuracy. Currently, YOLOv3 uses a Non-Max Suppression algorithm to eliminate multiple detections of the same object. This algorithm is picking the most confident overlaying box on any object to present it as the final prediction. This paper presents a new algorithm called Weighted Centers of Confidence Selection that increases the precision using a confidence-weighted average bounding box as a replacement to the existing bounding boxes without making any changes to the YOLOv3 convolutional neural network. We demonstrate how this algorithm works and compare its results to the results achieved by the YOLO's Non-Max Suppression algorithm, focusing on precision and achieving almost the same frame-speed as the original YOLOv3. This new approach allowed us to improve the average accuracy on the COCO dataset in comparison to the original YOLO's Non-Max Suppression.**

Keywords—YOLO, object detection, object localization, non-max suppression, weighted centers of confidence selection.

I. Introduction

You Only Look Once (YOLO) is the object detection algorithm first introduced by J. Redmon at al. in 2015 [1]. It has been updated several times [2] and [3]. The recent version YOLOv3 [3] uses a specific deep convolutional neural network Darknet that typically outputs many bounding box predictions for every image according to the grid resolution and the number of the used anchors. So, we end up with many inaccurate and unwanted bounding boxes that are many times associated with the same objects. We need to winnow such predictions to get a single, most precise prediction for each object. The YOLO uses the Non-Max Suppression algorithm [4], which finds the highest confidence bounding box predictions and removes the other less confident partially overlapping bounding boxes on the same image. With the leftovers, the bounding boxes around detected objects can be drawn.

In this paper, we propose a Weighted Centers of Confidence Selection algorithm that can replace the Non-Max Suppression algorithm for more precise predictions of the final bounding boxes. The presented algorithm takes the best overlapping bounding boxes on the same object based on their prediction confidences and calculates a new bounding box on top of the object. The proposed algorithm does not burden the whole process by extra time, leaving the algorithm to work still very fast. It only raises the precisions of the predicted bounding boxes in comparison to the YOLO's original Non-Max Suppression algorithm.

We present the comparisons of these two algorithms on the COCO dataset showing that the introduced algorithm does not require much more computations than a Non-Max Suppression algorithm and returns better predictions of bounding boxes on average. The introduced algorithm does not change anything in the YOLO convolutional neural network. It is focussed on the post-processing of the bounding boxes returned by the convolutional network.

II. Object Detection Algorithms

Object detection is one of the major visual tasks performed by a human visual system. It is a domain that has benefited immensely from the recent developments in deep learning. Object detection systems repurpose classifiers to perform detection using a sliding window approach and run at an evenly spaced location over the entire image like DPM [6]. Other one-stage or two-state detection approaches like R-CNN, Fast and Faster R-CNN [7, 8, 12], SSD [9], or RetinaNet [10, 11] use methods for proposals of regions and bounding boxes, run a classifier to refine these boxes and eliminate duplicate detections and rescore the boxes based on other objects in the image. Such approaches are complex and hard to optimize because of many pipelines and components that must be trained separately. Object detection pipelines generally extract a set of robust features from input images to identify objects in the feature space by classifiers or localizers [5, 13], which run either in sliding window fashion over the whole image or on some subset of regions in the image.

YOLO [1] introduced the object detection system that requires to solve a single regression problem predicting object classes, their location in the image, and bounding boxes. YOLO is a general-purpose detector that learns to detect a variety of objects of different classes simultaneously. Thanks to the single pipeline, it can achieve the speed of tens of frames per second that allow processing streaming video in real-time with less than 25 ms of latency. Nevertheless, the development of faster and more accurate YOLO versions [2, 3] it still lags behind state-of-the-art detection system the precision of prediction of the bounding boxes. This paper introduces a Weighted Centers of Confidence Selection algorithm to help to fix this problem.

This paper is focused on the YOLOv3 [3] (Fig. 1.) one-stage object detection and classification algorithm that is very fast and can predict bounding boxes and class probabilities directly from full images in one evaluation by a single neural network. YOLO’s detection process can be described in three stages. In the first stage, the convolutional neural network predicts many bounding boxes with different predicted confidences (Fig. 1.1). Next, the bounding boxes are filtered due to their confidences, so there are left only those with the confidences larger than a given threshold (Fig. 1.2). Finally, the Non-Max Suppression algorithm searches for the subset of the bounding boxes with the largest confidences which do not overlap too much.

The YOLO network can be optimized end-to-end directly on detection performance. Thanks to its simplicity, it outperforms many other detection algorithms like DPM [6], R-CNN [7], Faster R-CNN [12] with ResNet, and SSD [9] taking into account the processing time or the number of frames per second and the number of the detected classes of objects. To raise YOLO’s v3 performance [3] on the objects of different sizes, it predicts three different scales, having strides 32, 16, 8, respectively. It makes detections on scales 13 x 13, 26 x 26, and 52 x 52 with an input of 416 x 416.

The substantial part of the YOLO is the definition of the bounding boxes (b) (Fig. 2) which are predicted using their midpoints (b_x, b_y), their widths (b_w), and heights (b_h):

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w \cdot e^{t_w} \tag{3}$$

$$b_h = p_h \cdot e^{t_h} \tag{4}$$

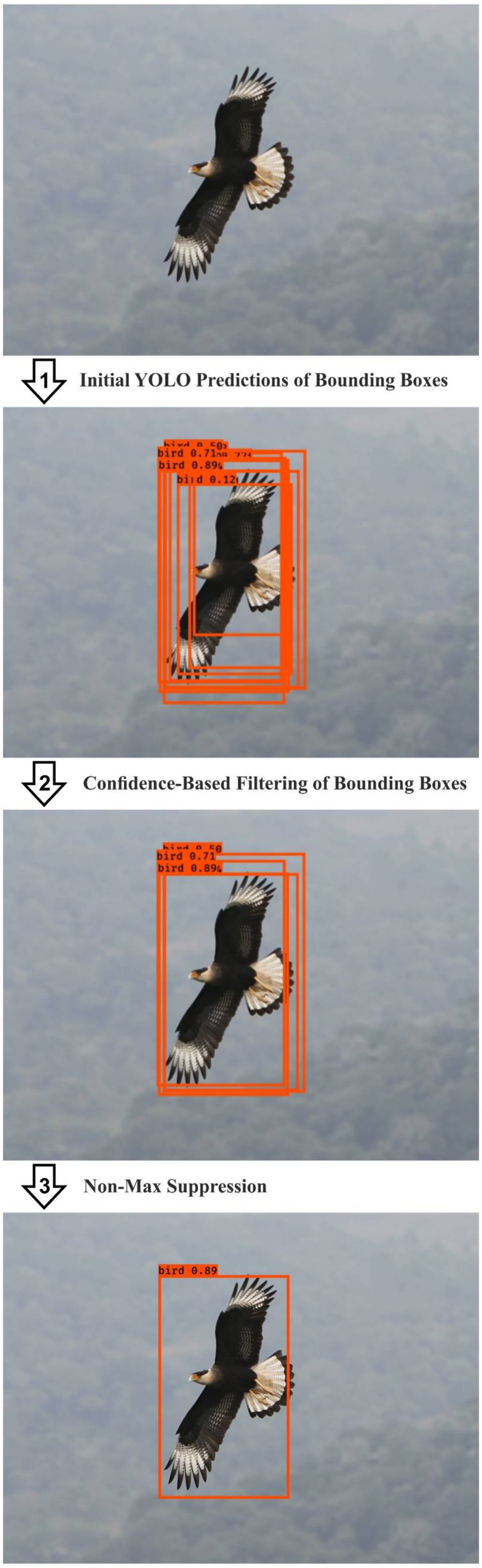

Fig. 1. Illustration of the YOLO process of prediction of bounding boxes.

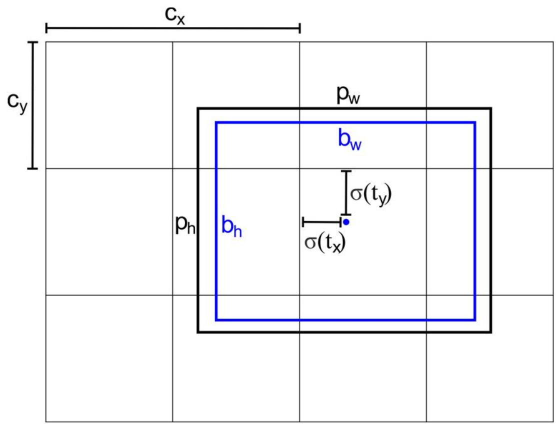

Fig. 2. Bounding boxes with the dimension and location prediction.

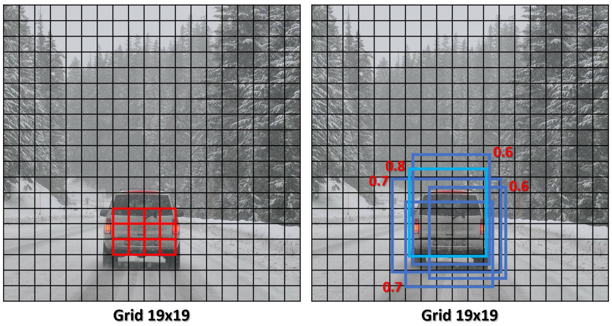

Fig. 3. Idea of Non-Max Suppression with the high-lighted grid cells (in red on the left) where the object was detected with high enough confidence and bounding boxes (on the right) which confidences were bigger than the minimum (threshold) confidence, from which the bounding box with the largest confidence is chosen by this algorithm as a final detection box.

where t_x, t_y, t_w, t_h is what the YOLO network outputs, c_x and c_y are the top-left coordinates of the grid cell, and p_w and p_h are the anchors' dimensions for the grid cell (box).

In YOLO, we use an objectness score p_o which is the probability that the midpoint of an object o is contained inside a grid cell. It is nearly 1 for the bounding boxes which well-bound an object, whereas almost 0 for the grid cells that do not contain any object of the defined classes.

YOLO divides the input image into a grid consisting of cells that are responsible for detecting objects when the midpoint of an object falls into a grid cell. Each cell predicts bounding boxes and confidence scores for those boxes. The confidence of the prediction is defined after [1] as:

$$c_o = p(o) \cdot IOU(b, o) \tag{5}$$

where the confidence should be zero when there is no object in that cell because of the probability $p(o)$ of the object is equal to zero in this case. Otherwise, the confidence is equal to the intersection over union (IOU) [1] between the predicted bounding box b and the ground truth (the box of the object o).

Each bounding box consists of 5 predictions representing the midpoint (b_x, b_y), width b_w, and height b_h of the bounding box of the detected object, as well as the probability $p(C_i|o)$ of the class C_i for this detected object o. The class-specific confidence scores for each box is computed as the product of the probabilities of $p(C_i|o)$ and the object detection $p(o)$ and the intersection over union IOU that collapses to the following formula:

$$s(o) = p(C_i|o) \cdot p(o) \cdot IOU(b, o) = p(C_i) \cdot IOU(b, o) \tag{6}$$

where $p(C_i|o)$ is the conditional class probability when detecting object o, and $p(C_i)$ is the class detection probability.

III. Non-Max Suppression

The Non-Max Suppression algorithm (NMS) [1, 2] is one of the key elements of the YOLO algorithm. It avoids multiple bounding boxes for the detected objects, leaving only one with the highest IOU. The problem of multiple classifications comes from the grid in which cells might think that they found an object of a given class and represent its midpoint. As a result, every such cell produces a bounding box, and thus we usually get multiple bounding boxes for the same object (Figs. 3 and 4.).

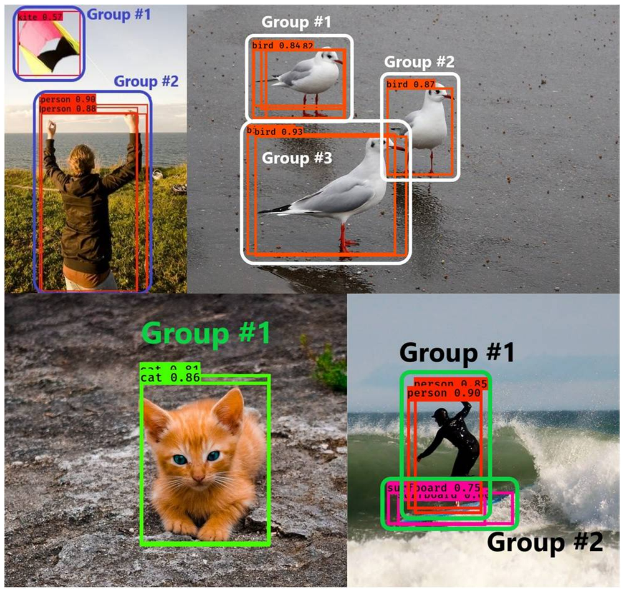

Fig. 4. Descriptive illustration of the grouping operation

YOLO's Non-Max Suppression is devoted to choose only one bounding box with the highest prediction confidence p_c computed for each object detection by grid cells. This algorithm selects the bounding box with the highest confidence for each detected object (from a group of bounding boxes (Fig. 4) which substantially overlap) but does not use the less confident bounding boxes or their predicted confidences any more. The

Non-Max Suppression Algorithm

Input: 3D matrix B with estimated bounding boxes produced by the convolutional YOLOv3 network
Parameters: Minimum accepted probability p_c^{min}, minimum $\mathrm{IOU}^{\mathrm{min}}$ above which bounding boxes are removed.
Output: Set L of bounding boxes for objects with the largest predictions.

```
B = set of bounding boxes
foreach b in B
    if b. p_c ≤ p_c^min then B.Remove(b)
end foreach
while B.IsNotEmpty do
    b = GetBoundingBoxWithMax p_c
    L.Add(b)
    foreach b_i in B
        if IOU(b, b_i) ≥ IOU^min then
            B.Remove(b_i)
        end if
    end foreach
end while
return L
```

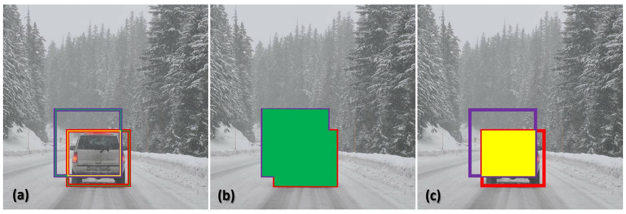

Fig. 5. Intersection over union (IOU) with (b) the presentation of the union and (c) the intersection.

less confident bounding boxes also carry useful information that can be used to compute the final bounding box for each object instead of selecting the one with the highest confidence only.

We use $IOU(b, b_i)$ (Fig. 5) that stands for the intersection over the union of the selected bounding box b with the largest confidence and other compared bounding boxes b_i which were not yet removed from the set of bounding boxes B. The $IOU(b, b_i)$ is computed as a ratio of the intersection size to the union size of two specified boxes. The result is always between 0 and 1, where 1 corresponds to the perfect overlapping of the same sized boxes, and 0 means boxes with no overlapping area.

From the theoretical point of view, the Non-Max Suppression algorithm removes possibly helpful but relatively less confident bounding boxes and loses their found predictions. These removed predictions may sometimes have only a tiny less confidence in comparison to the winning prediction bounding box. The general idea of our research was to use the less confident predictions instead of just removing them, and the results show that this approach helped to achieve generally better average precision of the detected objects.

IV. Weighted Centers of Confidence Selection

The Weighted Centers of Confidence Selection (WCCS) is an algorithm devoted to the post-processing part of the YOLOv3 to achieve more accurate predictions, not losing its performance. It computes the final bounding box for each object instead of selecting the one with the maximum confidence. It can be used as an alternative to Non-Max Suppression.

It starts from removing all bounding boxes with low prediction confidences (which are under a given threshold $p_c < p_c^\theta$) in a similar way as the Non-Max Suppression. Thus, it leaves a group the most confident bounding boxes $\{b_1, \ldots, b_n\}$ which $p_c \geq p_c^\theta$, e.g. $p_c^\theta = 0.4$. For the overlapping enough bounding boxes b_i and b_j with the confidence bigger than the threshold p_c^θ and for which $IOU(b_i, b_j) \geq \mathrm{IOU}^{Min\theta}$, e.g. $\mathrm{IOU}^\theta = 0.5$, we compute the weighted averages of their midpoints, widths, and heights (7) where weights are defined using the squared prediction confidences of the bounding boxes. $IOU(b_i, b_j)$ stands for the Intersection Over Union between the two bounding boxes b_i and b_j. In WCCS, the IOU is computed for every predicted bounding box which $p_c \geq p_c^\theta$. During the grouping process, it also filters out the predicted classes $\{C_1, \ldots, C_L\}$ of the object, so WCCS does not store bounding boxes of different classes in the same group.

$$\hat{b} = (\mathrm{b_x}, \mathrm{b_y}, \mathrm{b_w}, \mathrm{b_h}) = \begin{pmatrix} \dfrac{\sum_{k=1}^{n} \left(b_x^{(k)} \cdot \mathrm{c}_k{}^2\right)}{c_o^2} \\ \dfrac{\sum_{k=1}^{n} \left(b_y^{(k)} \cdot \mathrm{c}_k{}^2\right)}{c_o^2} \\ \dfrac{\sum_{k=1}^{n} \left(b_w^{(k)} \cdot \mathrm{c}_k{}^2\right)}{c_o^2} \\ \dfrac{\sum_{k=1}^{n} \left(b_h^{(k)} \cdot \mathrm{c}_k{}^2\right)}{c_o^2} \end{pmatrix} \tag{7}$$

$$c_o^2 = \sum_{k=1}^{n} c_o^{(k)^2} \tag{8}$$

After grouping is finished, it calculates a new bounding box $\hat{b}$ (7) using up to three most confident and overlapping bounding boxes from each group. The new bounding box is calculated based on the weighted averages where weights are the squared prediction confidences of those bounding boxes. Next, all of the bounding boxes in the group are removed, and only the calculated bounding box is kept. The following pseudo-code can provide more precise insight.

Grouping Algorithm using $IOU(b_i, b_j) \geq \mathrm{IOU}^{Min\theta}$

Input: 3D matrix defining the set of bounding boxes B produced by the convolutional YOLOv3 network
Parameters: Minimum threshold $\mathrm{IOU}^{Min\theta}$ of acceptance the overlapping bounding boxes.
Output: 2D matrix designed to define the set of groups G.

```
BNo = SizeOf(B)  # Number of bounding boxes in B
G = MatrixOfZeros(size=( BNo x BNo))
foreach b1 in B
    foreach b2 in B
        if b1 != b2 then
            if b1.class == b2.class then
                if IOU(b1, b2) ≥ IOU^Minθ then
                    G[b1, b2] = 1
                end if
            end if
        end if
    end foreach
end foreach
G.RemoveDuplicates()
return G
```

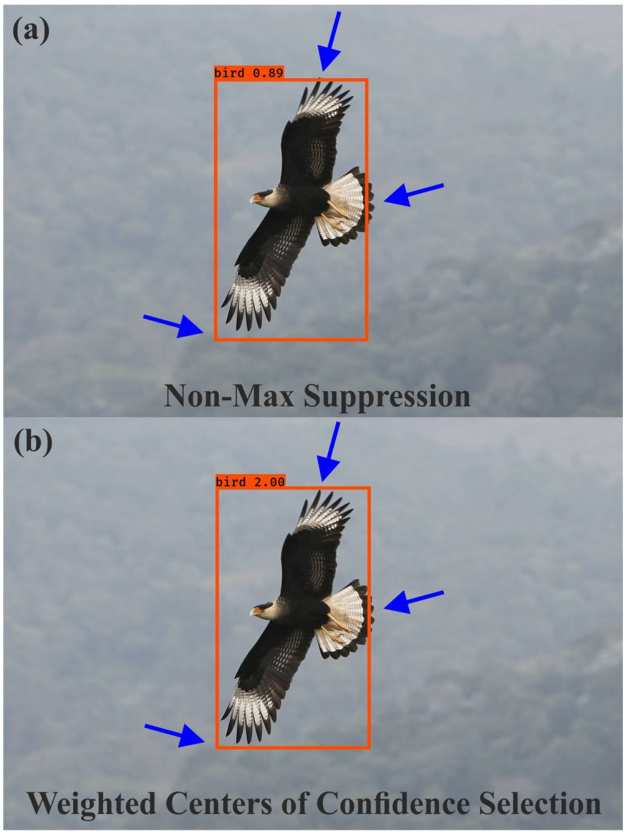

Fig. 6. Illustration of the results got using (a) Non-Max Suppressio, (b) Weighted Centers of Confidence Selections.

In Fig. 6, there is depicted the difference between the proposed and Non-Max Suppression algorithms. The top image shows the result of the Non-Max Suppression and the prediction of the selected bounding box. The bottom image presents the result of the Weighted Centers of Confidence Selection that is more precise than the previous one.

Due to not changing the YOLO's network, the root prediction accuracy has not risen, and it does not damage the time-efficiency of the improved algorithm. We merely adjust the way it processes after the prediction. Due to that, using commonly used [3] mAP would not be fair for this. Instead, we approached another way of checking the precision of bounding box predictions using an absolute error e (9) [14] comparing the ground truth bounding boxes $g^o = (g_x^o, g_y^o, g_w^o, g_h^o)$ with the predicted bounding boxes for each object o:

$$e = |\mathrm{b}_x - g_x^o| + |\mathrm{b}_y - g_y^o| + |\mathrm{b}_w - g_w^o| + |\mathrm{b}_h - g_h^o| \quad (9)$$

Weighted Centers of Confidence Selection

Input: 3D matrix B with estimated bounding boxes produced by the convolutional YOLOv3 network and the group matrix G from the Grouping Algorithm

Parameters: maxBBoxNo initialized to 3 defining the maximum number of bounding boxes of the largest prediction confidence that are used to compute the new bounding box by this algorithm.

Output: Set L of the computed bounding boxes for the groups of bounding boxes stored in matrix G.

```
B = set of bounding boxes
G = set of groups of bounding boxes
maxBBoxNo = 3
foreach g in G
    if g.Count() == 1 then
        L.Add(g.First())
    else if
        bb = new Bounding Box
        c2 = 0
        foreach b in g
            b2 = b.c * b.c
            c2 += b2
            bb.bx += b.bx * b2
            bb.by += b.by * b2
            bb.bw += b.bw * b2
            bb.bh += b.bh * b2
        end foreach
        bb.bx /= c2
        bb.by /= c2
        bb.bw /= c2
        bb.bh /= c2
        L.Add(bb)
    end if
    g.RemoveAllBoundingBoxes()
end foreach
return L
```

We tested this approach on the COCO dataset [5], comparing the Non-Max Suppression to the proposed algorithm using the ground truth bounding boxes provided by the COCO dataset. Below, the previously mentioned algorithm is presented with the following pseudo-code:

Absolute Error Calculation

Input: COCO dataset with the defined ground truth bounding boxes and classes for all objects of all images
Parameters: Threshold IOU^{θ} of acceptance the overlapping bounding boxes
Output: Absolute total error e.

```
e = 0
foreach image in COCO
    foreach pred_box in all_predictions[image.id]
        foreach coco_box in COCO[image.id].bbox
            if IOU(pred_box, coco_box) ≥ IOU^θ and
                pred_box.class == coco_box.class
            then
                e +=sum(difference(pred_box,coco_box))
            end if
        end foreach
    end foreach
end while
return e
```

The IOU^{θ} is a threshold that is for matching the exact boxes in order to compare precision, so usually $\mathrm{IOU}^{\theta} > \mathrm{IOU}^{Min\theta}$, where $\mathrm{IOU}^{Min\theta}$ was used in the Grouping Algorithm.

After storing the absolute errors with respect to the number of predictions made, we divide the absolute errors due to the number of predictions to get an overall average absolute error score presented for many predictions in Fig. 7.

V. Comparisons and Results

We used the COCO dataset that is typically used in comparisons of various object detection methods. During our tests, we used the entire validation part of the current version of the COCO dataset. We measured the average error on both approaches for all objects in all images. The results presented in Fig. 7 show that the improvement achieved by the use of WCCS is stable in comparison to the NMS; nevertheless, the number of predictions we made, so it proves that the replacement of NMS by WCCS has sense. As shown in Fig. 7, every bar represents the average absolute error for the growing number of predictions. For all presented averages, the presented approach using WCCS instead of NMS is winning in the precision of the predictions of the bounding boxes, and the improvement is stable so that we can use it without any doubt.

Comparing to the Non-Max Suppression, when the Weighted Centers of Confidence Selections algorithm is applied to images with objects on the COCO dataset, the absolute error is reduced by 2% on average. To show that this 2% improvement makes sense, some examples of such improvements are presented in Fig. 8.

We also measured the time efficiency of this new approach. It turned out that it did not burden the calculation time significantly. The comparison of the average YOLO prediction based on NMS to the average prediction made by YOLO with implemented WCCS showed that the time-efficiency decreased only about 0.27%. Hence, it is not meaningful from the frame per second processing speed of input images because we can lose at most one frame per second.

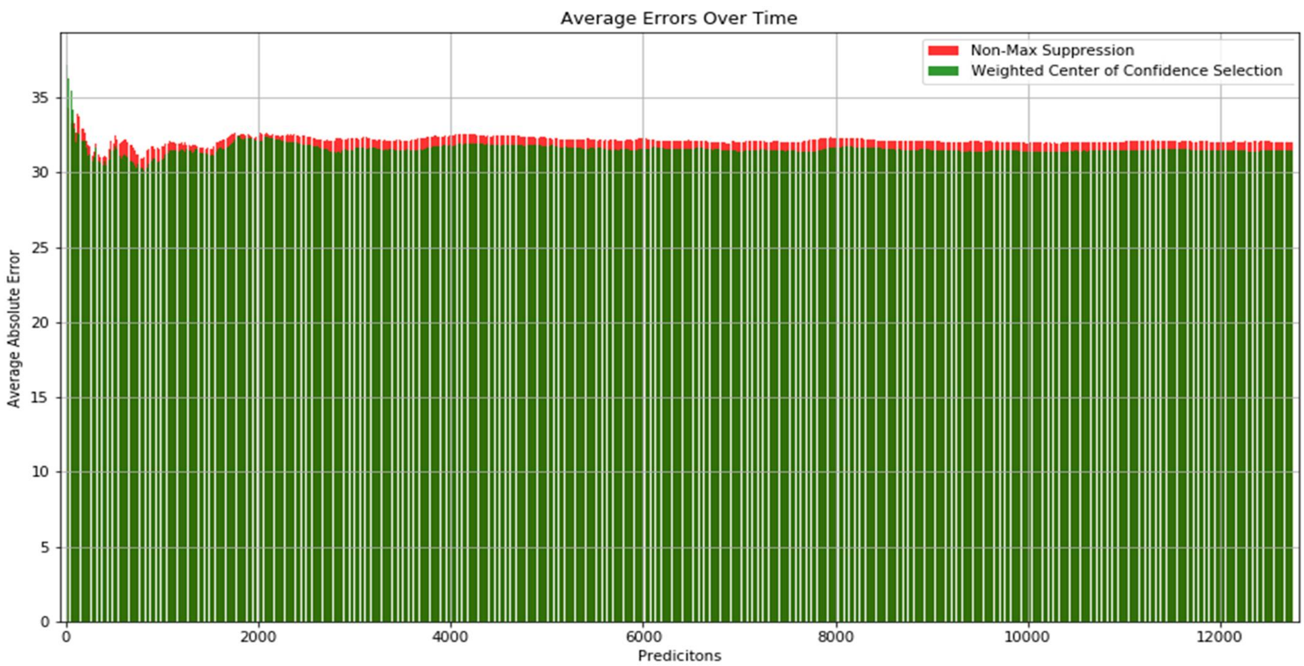

Fig. 7. Comparison of Average Absolute Errors achieved for Non-Max Suppression and Weighted Centers of Confidence Selection. The difference show us the better peformance on the WCCS algorithm.

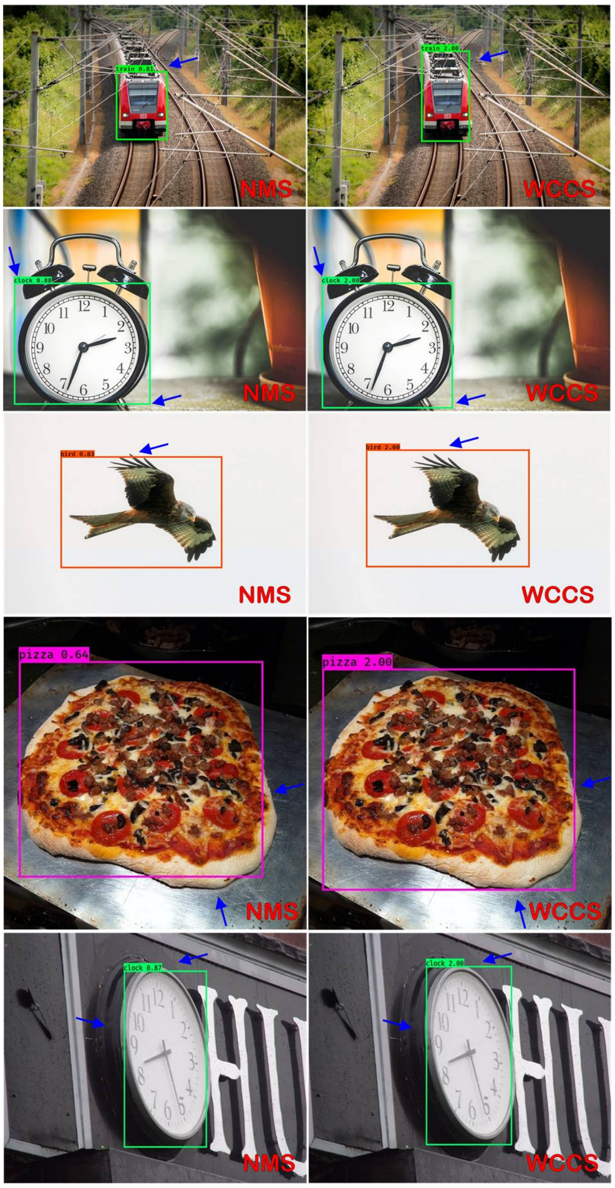

Fig. 8. Comparison of Non-Max Suppression (on the left in pairs) to Weighted Centers of Confidence Selection (on the right in pairs).

Once we categorize detected objects by their relative sizes regarding the entire picture in which they belong, we see that the method we introduce is performing dramatically better on the larger objects. In Table I and Fig. 9, we compare the objects of

TABLE I. COMPARISON OF MEAN ABSOLUTE ERROS OF THE OBJECTS OF DIFFERENT SIZE

Mean Absolute Error	**Object Size**		
	Small	***Midsize***	***Large***
NMS	21.5537	53.0042	78.0070
WCCS	21.3142	51.8339	75.5162

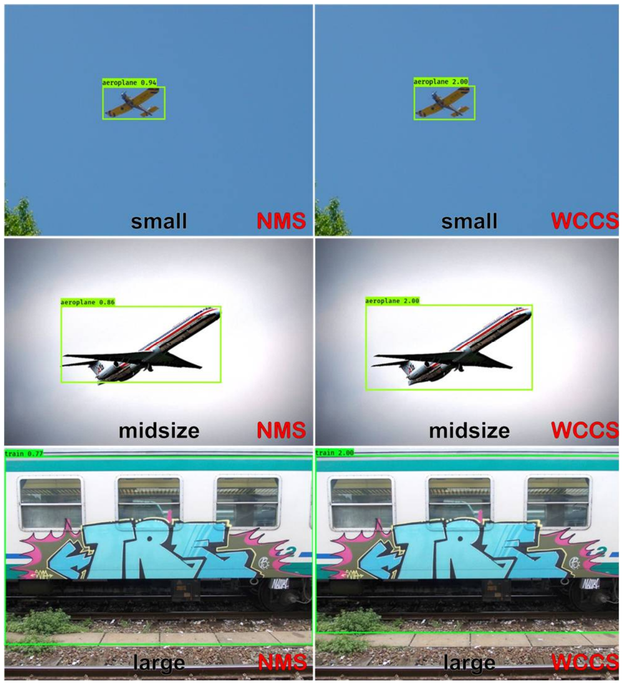

Fig. 9. Illustration of the sample results for (a) small, (b) midsize, and (c) large images using NMS (left column) and WCCS (right column).

three different groups based on their sizes: small (that cover less than 10% of the image), midsize (that cover between 10% and 45% of the image) and large (that cover more than 45% of the image) objects.

After the definition, the recall is the ratio of the number of the true object detections to the total number of objects in the data set. The average recall values and the average precision values with respect to the IOU threshold are presented in Figs. 10. and 11. The mean average precision (mAP) score (Fig. 11) is increased by 1.3%, and the mean average recall score is increased by 2.6% on average through the different levels of

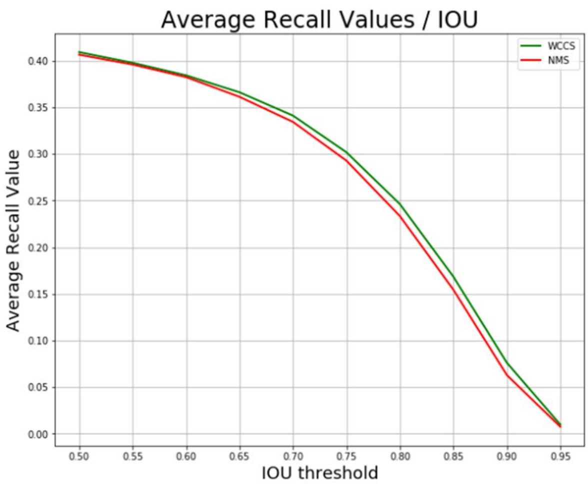

Fig. 10. The Average Recall Value with respect to the IOU threshold.

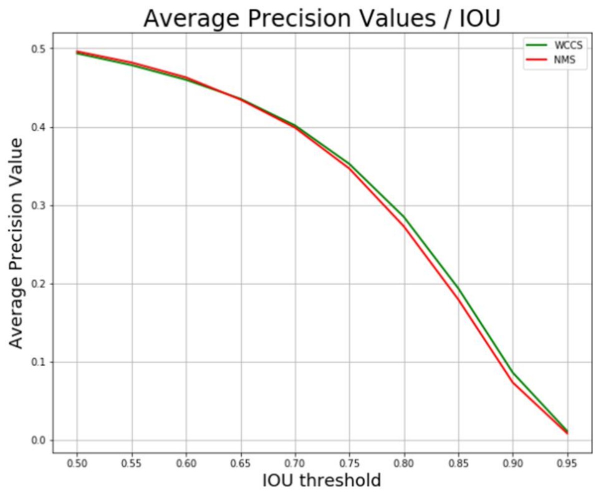

Fig. 11. The Average Precision Value with respect to the IOU threshold.

IOU thresholds. The 1.3% increase on average in the traditional mAP per IOU score in Fig. 10. However, as we mentioned previously in this paper, comparing Non-Max Suppression with Weighted Centers of Confidence by using mAP should not be recognized as the most important comparison in the paper. As the researcher who wrote this paper, we believe the proposed absolute error (9) is a better approach for understanding our post-processing improvement.

VI. Conclusion and Remarks

In this paper, we presented the improved post-processing algorithm of final bounding boxes calculations for YOLOv3. It has increased the precision of these bounding boxes in comparison to the original Non-Max Suppression. The proposed algorithm improved the accuracy of the predicted bounding boxes about 2,0% on average on the COCO dataset. It might be especially important for self-driving cars or cancer detection on X-Ray images and in many other applications. Every improvement could set a life and death difference or decide about road safety.

Moreover, the presented improvement does not change the famous YOLO's high performance because the proposed mathematical operations at the post-processing phase of the prediction are quite simple and can be quickly computed in comparison to the evaluation of the input images by the YOLO convolutional network. The costliest operation from the entire process of object detection is the prediction part made by this network. Therefore, the difference (0.27%) in computational time introduced by the proposed algorithm is unnoticeable.

The replacement of the Non-Max Suppression with the proposed Weighted Centers of Confidence Selection implemented to YOLOv3 allows for raising the average precision about 2% on the COCO dataset without changing the convolutional network of YOLOv3.

Acknowledgment

This work was supported by AGH 16.16.120.773.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, Real-Time Object Detection," arXiv preprint arXiv:1506.02640, 2015.

[2] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Xpress, 2017, pp. 6517–6525.

[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv, 2018.

[4] Alexander Neubeck and Luc Van Gool, "Efficient Non-Maximum Suppression," in 18th International Conference on Pattern Recognition (ICPR'06), vol. 3., IEEE, 2006.

[5] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," In European Conference on Computer Vision. Springer, 2014, pp. 740–755.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, issue 9, 2010, pp. 1627–1645.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in 2014 IEEE Computer Vision and Pattern Recognition (CVPR), IEEE Xpress, 2014, pp. 580–587.

[8] R. B. Girshick, Fast R-CNN," CoRR, abs/1504.08083, 2015.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," in European Conference on Computer Vision, pp. 21–37.

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar, "Focal Loss for Dense Object Detection," Facebook AI Research (FAIR), Cornell University, arXiv:1708.02002, 2017.

[11] Yixing Li, Fengbo Ren, Light-Weight, "RetinaNet for Object Detection," in IEEE Conference on Computer Vision and Pattern Recognition, Cornell University arXiv:1905.10011, 2019.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 39, issue 6, June 2017, pp. 1137–1149.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778, DOI: 10.1109/TPAMI.2016.2577031.

[14] Willmott, J. Cort, and Kenji Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," Climate research, vol. 30, issue 1, 2005, pp. 79–82.