Detecting Text on Historical Maps by Selecting Best Candidates of Deep Neural Networks Output

Gerasimos Matidis[1,2](✉), Basilis Gatos[1], Anastasios L. Kesidis[2], and Panagiotis Kaddas[1,3]

[1] Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,
NCSR "Demokritos",, 15310 Athens, Greece
{gmatidis,bgat,pkaddas}@iit.demokritos.gr

[2] Department of Surveying and Geoinformatics Engineering, University of West Attica,
12243 Athens, Greece
akesidis@uniwa.gr

[3] Department of Informatics and Telecommunications, University of Athens,
15784 Athens, Greece

Abstract. The final and perhaps the most crucial step in Object Detection is the selection of the best candidates out of all the proposed regions a framework outputs. Typically, Non-Maximum Suppression approaches (NMS) are employed to tackle this problem. The standard NMS relies exclusively on the confidence scores, as it selects the bounding box with the highest score within a cluster of boxes determined by a relatively high Intersection over Union (IoU) between each other, and then suppresses the remaining ones. On the other hand, algorithms like Confluence determine clusters of bounding boxes according to the proximity between them and select as best the box that is closer to the other ones within each cluster. In this work, we combine these methods by creating clusters of high confidence scores according to their IoU and then we calculate the sums of the Manhattan distances between the vertices of each box and all the others, in order to finally select the one with the minimum overall distance. Our results are compared with the standard NMS and the Locality-Aware NMS (LANMS), an algorithm that is widely used in Object Detection and merges the boxes row by row. The research field that this work explores is the text detection on historical maps and the proposed approach results to average precision that is 2.14–2.94% higher for evaluation IoU in range 0.50 to 0.95 with step 0.05 than the two other methods.

Keywords: Text Detection · Historical Maps · Non-Maximum Suppression

1 Introduction

Historical maps are an important and unique source of information for studying geographical transformations over years. In this paper, the focus is on the text detection task of the digital historical map processing workflow. This task is very important and

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
G. A. Fink et al. (Eds.): ICDAR 2023, LNCS 14191, pp. 358–367, 2023.
https://doi.org/10.1007/978-3-031-41734-4_22

crucial, since it provides the input for the recognition process that follows. At the same time, this task is extremely challenging due to the complex nature of the historical maps. As it can be observed in the example of Fig. 1, text in the historical maps can be of any size, any orientation, may be curved, with variable spacing and usually overlaps with other graphical map elements.

Current approaches for text detection in historical maps include the use of color and spatial image and text attributes [1, 2]. In [3], text in maps is identified based on the geometry of individual connected components without considering most of the aforementioned text detection challenges. Approach [4] uses 2-D Discrete Cosine Transformation coefficients and Support Vector Machines to classify the pixels of lines and characters on raster maps. Recently, Convolutional Neural Network (CNN) architectures have been proved efficient for text detection in historical maps. In [5] and [6], Deep CNNs are introduced for end-to-end text reading of historical maps. A text detection network predicts word bounding boxes at arbitrary orientations and scales. Several text detection neural network models are evaluated in [7] and [8]. The pixel-wise positions of text regions are detected in [9] by employing a CNN-based architecture.

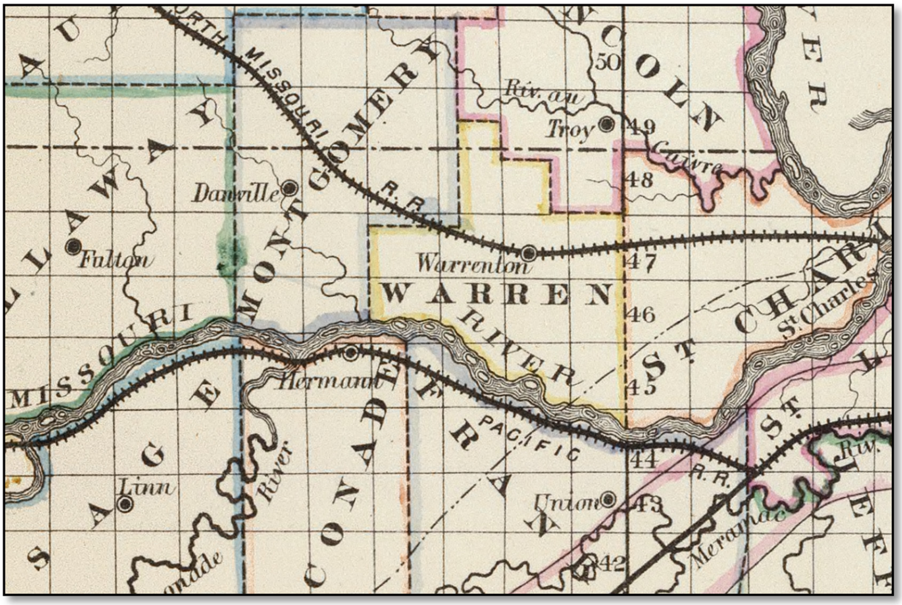
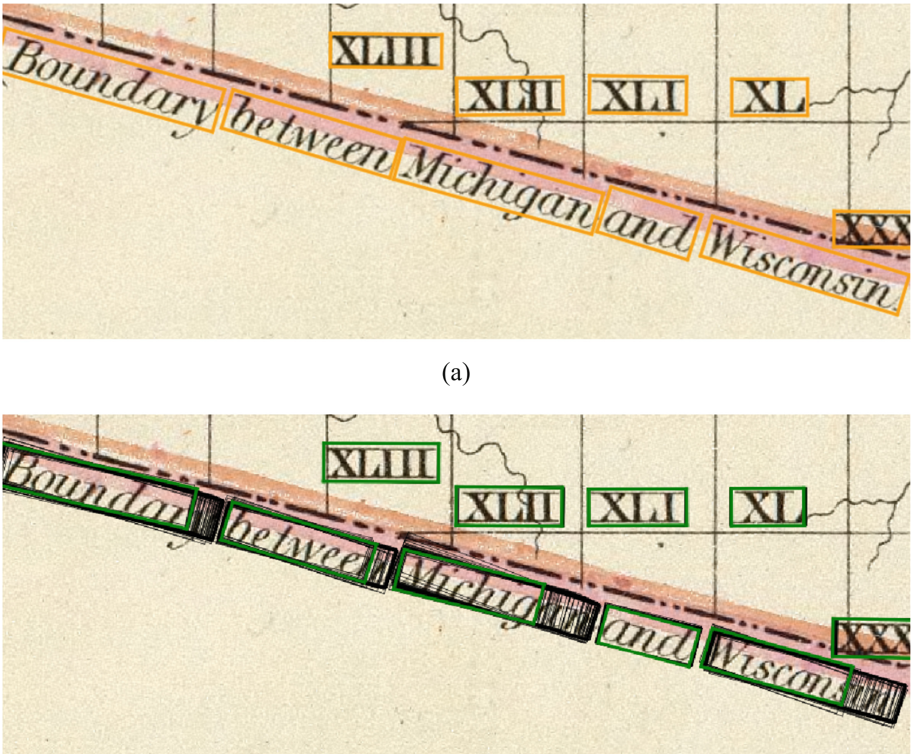


Fig. 1. Part of a historical map from the dataset provided in [5].

Following the recent promising approaches based on CNN architectures, in this work we first apply the deep neural network of [5] and focus on the final and perhaps the most crucial step for text detection, which is the selection of the best candidates out of all the proposed regions coming as output of the CNN framework. As it is demonstrated in Fig. 2, the network output usually corresponds to several overlapping blocks around the text area (Fig. 2b), while the desired final output is just one block around the text area (Fig. 2a). This is a common problem for object detection applications and several Non-Maximum Suppression (NMS) approaches has been proposed for solving it. The standard NMS relies exclusively on the confidence scores, as it selects the bounding box with the highest score within a cluster of boxes determined by a relatively high Intersection over Union (IoU) between them, and then suppresses the remaining ones. The Locality-Aware NMS (LANMS) [10], also used in [5], is based on merging the boxes row by row. On the other hand, algorithms like Confluence [11] determine clusters of bounding boxes according to the proximity between them and they select as best the box that is closer to the other ones within each cluster. However, it can be observed that in NMS-based approaches the most confident box does not always correspond to the best solution (Fig. 2b). Indeed, there may exist other candidate boxes, with slightly lower confidence score than the selected one, which provide more accurate predictions of the desired bounding box. In the proposed work, we try to find the best solution that can be applied to the difficult case of historical maps and combine the standard NMS with the Confluence approach. Initially, we create clusters of high confidence scores according to



(a)

(b)

Fig. 2. (a) The ground truth bounding boxes (orange), (b) The total high-confident boxes predicted by the network (black) and the ones with the maximum confidence score for each word (green). As it can be observed, the most confident bounding boxes do not always correspond to the best solution.

their IoU. Then, we calculate the sums of the Manhattan distances between the vertices of each box and all the others in the cluster, in order to finally select the one with the minimum overall distance. To achieve this, we also generalize the Confluence algorithm in order to process blocks of any orientation. As it is demonstrated in the experimental results, the proposed method gives an average boost of 2.14–2.94%, concerning the average precision metric for IoU in range 0.50 to 0.95 with step 0.05, when compared with standard NMS and LANMS.

The rest of the paper is organized as follows: Section 2 introduces the proposed text detection method, Section 3 demonstrates the experimental results and Sect. 4 presents the conclusion of this work.

2 Methodology

Recent Object Detection networks usually provide a very large number of bounding box predictions, each one assigned with a confidence score. The number ranges from a few thousand, in cases of region proposal-based [12] or grid-based [13] detectors, to millions, in cases of dense [10] detectors (one prediction per pixel).

As mentioned in the previous section, the standard NMS selects repetitively the bounding box with the maximum confidence score and suppresses the ones that significantly overlap with it. However, it appears that in practice this is not always the best strategy. In particular, in many cases the most confident box misses a considerable part of the text, while other ones capture it more precisely (Fig. 2b). A second issue is that many of the boxes, which are to be suppressed and they have a slightly lower confidence score than the most confident box, correspond to better predictions of the ground truth bounding box. This situation occurs especially when dealing with dense-detection networks. Indeed, in our experiments we reported various cases, where the difference in confidence between the most confident box and some of the suppressed boxes is less than 10^{-4}, while the later ones corresponded to better prediction accuracy.

Considering the above, we repetitively define clusters of boxes, which consist of the most confident and the ones that significantly overlap with it, such as the standard NMS. Then, by calculating the sums of the Manhattan distances between the vertices of each one and the other boxes in the cluster, the one with the minimum overall distance is selected, similarly to [11]. However, we extend [11] in order to include rotated boxes. The Manhattan distances are then calculated for all the four vertices of every pair of boxes, instead of two diagonal ones. Figure 3 depicts the process of the calculation of the distance between two boxes.

In this work, the MapTD network [5], which produces dense predictions, is used for detecting text on the maps. A threshold of 0.95 is applied in order to eliminate the bounding boxes with low confidence scores. Let B denote the list of the remaining candidate boxes. The main steps of the proposed algorithm are as follows:

Step 1: Sort the list B in a descending order with respect to the confidence scores of the candidate bounding boxes.

Step 2: Initiate an empty list F to store the final boxes.

Step 3: Select the first (most confident) box in B and calculate the IoU between this box and every other box in the list.

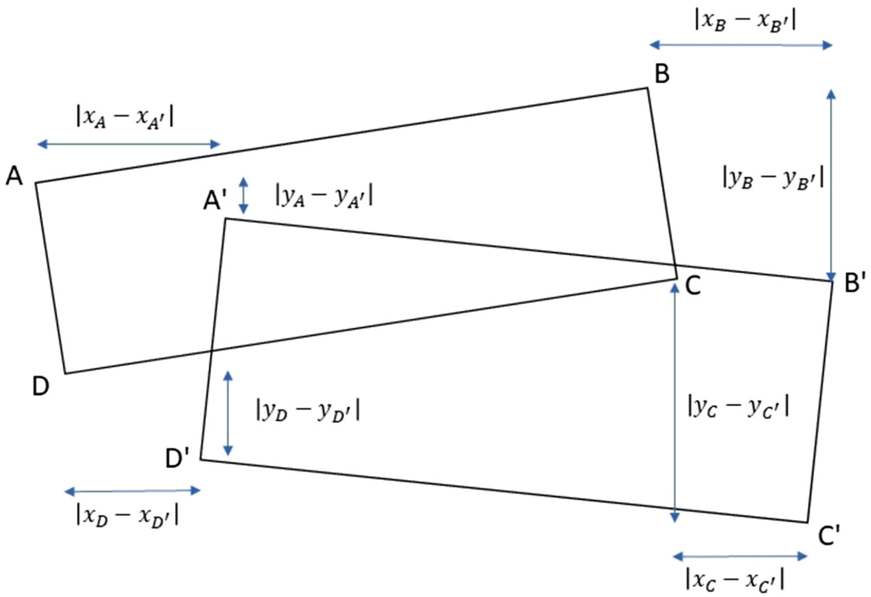


Fig. 3. Example of the distance calculation between two bounding boxes ABCD and A′B′C′D′. The distance is the sum of the Manhattan distance between every pair of similar vertices.

Step 4: Define a cluster by creating a list C including the most confident box and all the boxes that have a minimum overlap with it, which is determined by a predefined IoU threshold. Remove all the boxes that are stored in list C from the main list B.

Step 5: Calculate the distances between each box and every other box in list C. Specifically, t he distance between two bounding boxes b_i with vertices ABCD and b_j with vertices A′B′C′D′ as shown in Fig. 3, is given by calculating the sum of the Manhattan distances between the four pairs of the corresponding vertices. Following the approach described in Sect. 3.2 of [11], the coordinates are normalized as follows:

$$x_i = \frac{x_i - \min(X)}{\max(X) - \min(X)},\ y_i = \frac{y_i - \min(Y)}{\max(Y) - \min(Y)},\ \forall i \in \{A, B, C, D, A', B', C', D'\} \tag{1}$$

where

$$X = \{x_A, x_B, x_C, x_D, x_{A'}, x_{B'}, x_{C'}, x_D\}$$

and

$$Y = \{y_A, y_B, y_C, y_D, y_{A'}, y_{B'}, y_{C'}, y_{D'}\}$$

The distance between two bounding boxes is calculated as

$$\begin{aligned} D_{b_i,b_j} = {} & |x_A - x_{A'}| + |y_A - y_{A'}| + |x_B - x_{B'}| + |y_B - y_{B'}| + \\ & + |x_C - x_{C'}| + |y_C - y_{C'}| + |x_D - x_{D'}| + |y_D - y_{D'}| \end{aligned} \tag{2}$$

The overall distance D_b for every box in cluster C, is the sum of all the distances between bounding box b and all the other boxes as follows:

$$D_b = \sum_{i=1}^{N_C} D_{bb_i}, \forall b \in C, b \neq i \tag{3}$$

where N_C denotes the number of boxes in C.

Step 6: Weight the total distance D_b as:

$$D_b = D_b(1 - S_b + \varepsilon) \tag{4}$$

where $S_b \in [0, 1]$ is its confidence score and ε is a positive number near zero, which ensures that the total distance does not vanish when $S_b \rightarrow 1$. In our experiments, ε is set to 0.1.

Step 7: Select the box with the minimum D_b and store it in the list F.

Step 8: Repeat steps 3 to 7 until list B becomes empty.

3 Experimental Results

In order to evaluate the proposed method, we trained MapTD models using the same dataset and training details as [5]. The dataset consists of 31 historical maps of the USA from the period 1866–1927 and it is publicly available[1]. We also follow the same 10-fold cross-validation of [5]. In particular, in every fold we use 27 maps for training and 3 maps for testing. One map is held out for validation across all folds. For the training we use Minibatch Stochastic Gradient Descent with minibatch size 16 × 512 × 512 and Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$. The learning rate is $\alpha = 10^{-4}$ for the first 2^{17} training steps and $\alpha = 10^{-5}$ for the rest of a total of 2^{20} training steps. For evaluation on each of the 3 test images of each fold, we take predictions on overlapping tiles of size 4096 × 4096, , with stride equal to 2048. The predicted bounding boxes of the network are filtered with a confidence score threshold of 0.95, in order to keep the most confident ones. All the three NMS methods use the same IoU threshold, equal to 0.1, in order to ensure that even slightly overlapping bounding boxes will belong to the same cluster [5].

Table 1 represents in details the average results across all folds for IoU values in range 0.50 to 0.95 with step 0.05. As it can be observed, the proposed algorithm outperforms the standard NMS and LANMS in average by 2.14% and 2.94%, respectively.

The differences between the proposed and the other two methods increase as the evaluation threshold becomes higher and reach a maximum when the threshold is 0.8. More specifically, the differences are as follows:

- Proposed – standard NMS &#xF0E0; **0.50**: 1.41%, **0.55**: 1.83%, **0.60**: 2.62%, **0.65**: 3.18%, **0.70**: 4.17%, **0.75**: 5.17%, **0.80**: 5.17%, **0.85**: 4.33%, **0.90**: 1.55%, **0.95**: 0.03%
- Proposed – LANMS &#xF0E0; **0.50**: 1.79%, **0.55**: 2.14%, **0.60**: 2.72%, **0.65**: 2.84%, **0.70**: 3.39%, **0.75**: 3.33%, **0.80**: 3.38%, **0.85**: 1.77%, **0.90**: 0.09%, **0.95**: -0.01%

[1] Https://weinman.cs.grinnell.edu/research/maps.shtml#data

Table 1. Overall Evaluation (Confidence threshold = 0.95, NMS threshold = 0.1)

Evaluation IoU threshold	Method	Predicted boxes	Ground Truth boxes	Correctly predicted boxes	Average Precision
0.50	standard NMS	32583	33315	29954	85.99%
	LANMS	32436		29955	85.61%
	Proposed	32583		30081	**87.40%**
0.55	standard NMS	32583	33315	29567	84.06%
	LANMS	32436		29625	83.75%
	Proposed	32583		29734	**85.89%**
0.60	standard NMS	32583	33315	29045	81.44%
	LANMS	32436		29163	81.33%
	Proposed	32583		29310	**84.06%**
0.65	standard NMS	32583	33315	28275	77.72%
	LANMS	32436		28493	78.07%
	Proposed	32583		28587	**80.90%**
0.70	standard NMS	32583	33315	27012	71.67%
	LANMS	32436		27289	72.45%
	Proposed	32583		27453	**75.83%**
0.75	standard NMS	32583	33315	24836	61.79%
	LANMS	32436		25306	63.62%
	Proposed	32583		25423	**66.96%**
0.80	standard NMS	32583	33315	21285	46.83%
	LANMS	32436		21752	48.62%
	Proposed	32583		21924	**52.00%**
0.85	standard NMS	32583	33315	15221	25.14%
	LANMS	32436		15917	27.70%
	Proposed	32583		15996	**29.47%**

(*continued*)

Table 1. (*continued*)

Evaluation IoU threshold	Method	Predicted boxes	Ground Truth boxes	Correctly predicted boxes	Average Precision
0.90	standard NMS	32583	33315	6880	5.45%
	LANMS	32436		7562	6.92%
	Proposed	32583		7573	**7.00%**
0.95	standard NMS	32583	33315	737	0.07%
	LANMS	32436		892	**0.11%**
	Proposed	32583		868	0.10%
Average results	standard NMS				54.02%
	LANMS				54.82%
	Proposed				**56.96%**

Figure 4 depicts two representative examples of how the proposed method results in more accurate bounding boxes. In analogy to the differences on the average precision, the bounding boxes of the proposed algorithm are better than these of LANMS, the boxes of which are better than standard NMS.

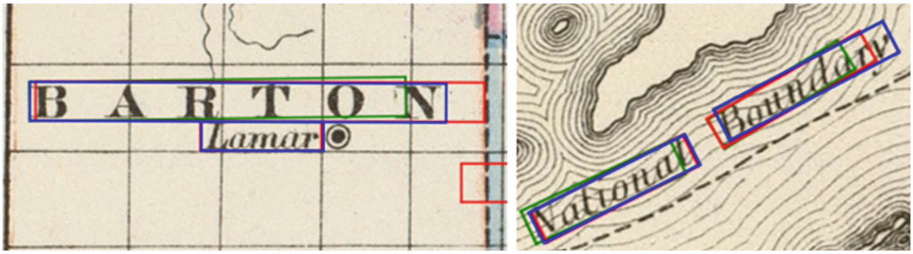


Fig. 4. Two samples with predicted bounding boxes for the three methods: standard NMS (green boxes), LANMS (red boxes) and proposed method (blue boxes). (Color figure online)

4 Conclusion

In this paper, a method is proposed that increases the accuracy of text detection on historical maps. It focuses on a particular post-processing step of the text detection pipeline by selecting the best solution among a large set of candidate bounding boxes, which are predicted by a deep CNN. To this direction, two existing Non-Maximum

Suppression methods are combined, namely, the standard NMS and the Confluence. The proposed method tackles a problem of NMS-based approaches, where the most confident box does not always correspond to the best solution, since there may exist other candidate boxes, with slightly lower confidence score than the selected one, which provide more accurate predictions of the desired bounding box. Instead of eliminating the bounding boxes that significantly overlap with the most confident, the method creates a cluster with all of them and selects as best the box that is closer to all others.

The experimental results show that the proposed method outperforms the standard NMS and LANMS on average precision metric and results to more accurate bounding boxes, a step that is very important, since it provides the input for the recognition process that follows text detection.

Acknowledgments. This research has been partially co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call “RESEARCH-CREATE-INNOVATE”, project Culdile (Cultural Dimensions of Deep Learning, project code: T1EΔK-03785) and the Operational Program Attica 2014–2020, under the call "RESEARCH AND INNOVATION PARTNERSHIPS IN THE REGION OF ATTICA", project reBook (Digital platform for re-publishing Historical Greek Books, project code: ATTP4-0331172).

References

1. Chiang, Y., Leyk, S., Knoblock, C.: A survey of digital map processing techniques. ACM Comput. Surv. **47**(1) (2014)
2. Velázquez, A., Levachkine, S.: Text/graphics separation and recognition in raster-scanned color cartographic maps. In: Lladós, J., Kwon, Y.-B. (eds.) GREC 2003. LNCS, vol. 3088, pp. 63–74. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25977-0_6
3. Pouderoux, J., Gonzato, J., Pereira A., Guitton, P.: Toponym recognition in scanned color topographic maps. In: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), pp. 531–535. Curitiba, Brazil (2007)
4. Chiang, Y.-Y., Knoblock, C.A.: Classification of line and character pixels on raster maps using discrete cosine transformation coefficients and support vector machine. In: 18th International Conference on Pattern Recognition (ICPR 2006), pp. 1034–1037. Hong Kong, China (2006)
5. Weinman, J., Chen, Z., Gafford, B., Gifford, N., Lamsal, A., Niehus-Staab, L.: Deep neural networks for text detection and recognition in historical maps. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 902–909. Sydney, NSW, Australia (2019)
6. Schlegel, I.: Automated extraction of labels from large-scale historical maps. AGILE: GIScience Series, vol. 2, pp. 1–14 (2021)
7. Lenc, L., Martínek, J., Baloun, J., Prantl, M., Král, P.: Historical map toponym extraction for efficient information retrieval. In: Uchida, S., Barney, E., Eglin, V. (eds.) Document Analysis Systems, DAS 2022, LNCS, vol. 13237, pp. 171–183. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06555-2_12
8. Philipp, J.N., Bryan, M.: Evaluation of CNN architectures for text detection in historical maps. In: Digital Access to Textual Cultural Heritage (DATeCH 2019) (2019)
9. Can, Y.S., Kabadayi, M.E.: Text detection and recognition by using cnns in the austro-hungarian historical military mapping survey. In: The 6th International Workshop on Historical Document Imaging and Processing (HIP 2021), pp. 25–30. Lausanne Switzerland (2021)

10. Zhou, X., et al.: EAST: an efficient and accurate scene text detector. In: The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) 2017, pp. 2642–2651. Honolulu, Hawaii (2017)
11. Andrew, S., Gregory, F., Paul, F.: Confluence: a robust non-IoU alternative to non-maxima suppression in object detection. IEEE Trans. Pattern Anal. Mach. Intell. (2021)
12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587. IEEE (2014)
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp. 779–788 (2016)