OBJECT LOCALIZATION BY OPTIMIZING CONVOLUTIONAL NEURAL NETWORK DETECTION SCORE USING GENERIC EDGE FEATURES

Elham Etemad, Qigang Gao

Dalhousie University
Faculty of Computer Science
Halifax, Canada

ABSTRACT

In this research, we propose an object localization method to boost the performance of current object recognition techniques. This method utilizes the image edge information as a clue to determine the location of the objects. The Generic Edge Tokens (GETs) of the image are extracted based on the perceptual organization elements of human vision. These edge tokens are parsed according to the Best First Search algorithm to fine-tune the location of objects, where the objective function is the detection score returned by the Deep Convolutional Neural Network. We have evaluated our method on top of the RCNN object recognition method. The results on Pascal VOC 2007 and 2012 show improved object localization performance. We also present several cases where the proposed method works significantly more precisely than RCNN.

Index Terms— Best First Search, Convolutional Neural Networks, Generic Edge Tokens, Object Localization, RCNN

1. INTRODUCTION

Object recognition is an essential technique in creating vision-aided agents and has been studied broadly since the 1960s [1]. From this extensive literature, the definition of an object is different based on the task to be tackled. In one definition, the object is defined as a set of templates that specify the features of the object in different conditions, while in another definition, when the task is more abstract, the object is defined by the contextual knowledge of the scene and is less dependent on a set of templates [2].

Considering the second definition of an object, researchers have conducted much research on object recognition, object detection, and object localization. In general, object detection determines whether an object exists in an input image, object recognition finds all the existing objects in the image, and object localization finds the precise location of those recognized objects [3].

To recognize objects in an image, researchers have introduced many methods to find the possible locations for objects called object proposal generation methods. Methods such as sliding window [4], selective search [5], hierarchical segmentation [6], and complexity adaptive [7] are among the techniques to find object proposals. When the proposals are generated, the area in each of them should be investigated to determine the existance of an object. The recent deep learning based methods targetting this problem include RCNN [8], Fast RCNN [9], Faster RCNN [10], and DeepID-Net [11]. The last of these utilized the fact that Deformable Part Models [12] are equivalent to Convolutional Neural Networks [13] and improved the object recognition performance.

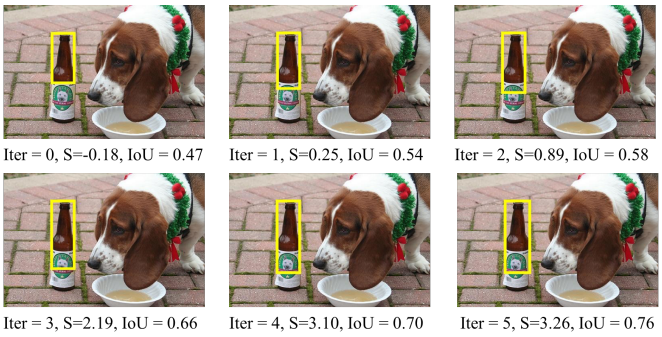


Fig. 1: Improved bounding boxes after several iterations of the Best First Search on GETs. The detection score and IoU with ground truth object has improved.

When the objects are recognized in an image, the precision of their locations should be improved by object localization techniques. The most popular object localization module is Bounding Box regression introduced by [8] which is a supervised method for improving the location of the recognized objects. Bilen [14] also introduced a method by defining an objective function like the Latent SVM and defining a similarity measure between the object window and its representative cluster. By identifying the superpixel tightness and straddling expansion,[15] expands the recognized bounding box to the tightest bounding box around the object.

In this research, an object localization method is introduced which relies on the fact that objects in the image have corresponding edge elements in the edge map of the image. This method applies a Best First Search algorithm [16] on the edge elements around the candidate objects, which are rec-

ognized by an object recognition module, one at a time. In each iteration, the current candidate object is merged with all its overlapping edge elements, and the CNN score for each merged box is calculated. The merged box with the maximum score is selected as an improved candidate object and is fed into the next iteration. This routine continues until there is no more edge or no more improvement.

The main difference of the proposed method from the current object localization methods such as bounding box regression [8] is its independence from any information about the training dataset. This means the proposed method solely relies on the information obtained from the current image. This feature creates the ability to apply the proposed method to the applications where a trained network exists, and the training data either does not exist or is expensive to obtain.

This paper is organized in the following order: The proposed method is explained and elaborated in Section. 2. The evaluation and discussion on the performance of the proposed method are illustrated in Section. 3. We conclude our paper in Section. 4 along with some possible areas of future work and the limitations of the proposed method.

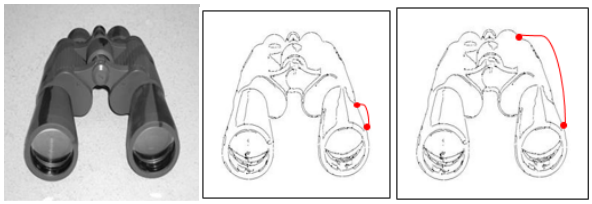
2. PROPOSED METHOD

The proposed object localization method comes after the object recognition method has found a set of candidate objects with their corresponding types and detection scores in the image. It modifies the locations of these candidate objects to improve their detection scores for their corresponding types, or for other types of objects with higher detection scores.

The proposed method applies a Best First Search on the set of edge elements, which represents the object boundaries in the image. The search space for this method is edge elements of the image extracted from its edge map. This method searches for the locations of objects in the image where the detection score of CNN is maximized. The improved candidate object in each iteration is a modified box whose detection score is higher than the original one, and is the input to the next iteration of the searches. This search continues until the improvement is stabilized or there is no more edge element with positive overlap with the current candidate box. Fig. 1 shows several iterations of GET_Loc and its positive impact on the object localization.

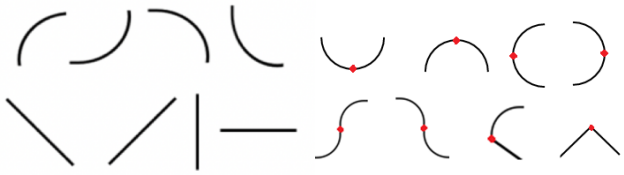
2.1. Edge Elements

The edge map of an image contains edge elements that are representatives of object boundaries in that image. (Fig. 2). This information is a useful source for improving the object localization, especially when the training images are not available. Using the PCPG package [17], the edge map of the image is obtained, a sample of which is represented in Fig. 2. This edge map is calculated by horizontal and vertical scanning of the input image and comparing the intensity values of



(a) Image (b) GET (c) Trace

Fig. 2: A sample Image (a) along with its edge map where a GET (b) and a Trace (c) are marked.



(a) GETs (b) CPPs

Fig. 3: (a) Eight types of GETs which consist of four categories of lines and four categories of curves. (b) Examples of Curve Partitioning Points (CPPs).

neighboring pixels, where the pixel whose gradient with its neighbor is higher than a threshold, represents an edge pixel.

The PCPG package recognizes the traces in the edge map by clustering its edge pixels based on their connectivity. Each trace is a set of edge pixels that are connected to each other and are separated from the other edge pixels, a sample of which is shown in Fig. 2c.

Each trace in the edge map consists of a set of Generic Edge Tokens (GETs) which are connected through Curve Partitioning Points (CPPs), represented in Fig. 3b. Each CPP is a corner on the edge, where the orientation and direction or both has changed. These GETs are classified based on their perceptual features into eight groups shown in Fig. 3a.

The traces and GETs are considered as edge elements that represent the object boundaries. Each of these elements locations in the image is represented by a bounding box which is used for finding the edge elements whose overlap with the candidate object is greater than zero, and for merging the edge elements with the candidate objects bounding box.

2.2. Optimization

For improving the precision of objects locations in the images, the Best First Search (BFS) algorithm has been utilized in this research. The main algorithm for the proposed method is represented in Algorithm 1. To adapt the BFS method into the object localization task, its search space, its objective function, and its finishing conditions are specified.

The search space for finding a more precise object location is a set of edge elements extracted from the image's edge map. In each iteration, the edge elements whose bounding boxes have overlap with the candidate object are selected as

Algorithm 1 Object localization using the GETs of the image

procedure GETLoc($Image, CanObj$)
  Input: Image
  Input: List of candidate boxes with their detection scores
  Output: List of recognized boxes with their detection scores
  for $Each\ CandidBox_i$ **do**
    while $Detection\ Score\ Improves$ **do**
      $FindMergedBoxes(CandidBox, EdgeMap)$
      for $Each\ Merged\ Box\ j$ **do**
        ▷ *Calculate Detection Score* $DS_{i,j}$
        $DS_{i,j} = CNNScore(MergedBox_j)$
      ▷ *Find the best merged box*
      $SelectedBox\ =\ arg\ max_{j \in MergedBox} DS_{i,j}$
      $CandidBox_i = SelectedBox$

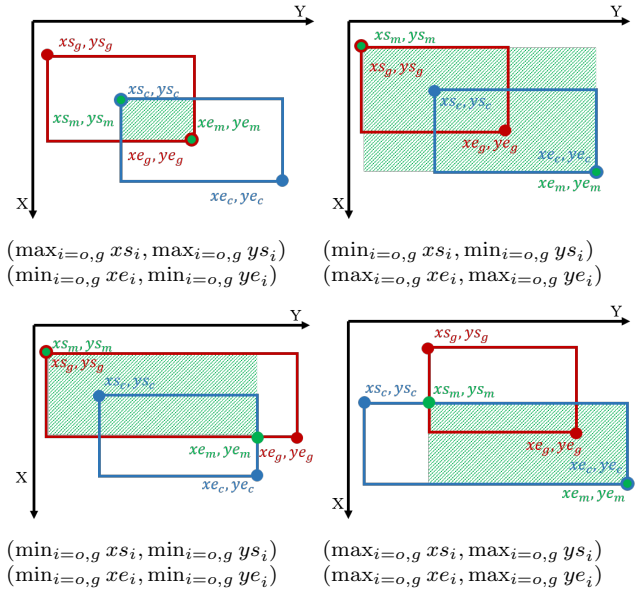


Fig. 4: Merge a candidate object with an overlapping edge element. The first and second rows represent (xs_m, ys_m) and (xe_m, ye_m) respectively.

possible improvements. This overlap is measured by Intersection over Union metric (IoU) between the candidate object's and the edge element's bounding boxes. In Fig. 4, a candidate object with coordination of $O = [xs_o, ys_o, xe_o, ye_o]$ is combined in four different ways with an edge element with a coordination of $G = [xs_g, ys_g, xe_g, ye_g]$ whose overlap with the candidate object is positive, and a set of merged boxes, $S_M = \{M_1, ..., M_n\}, M_i = [xs_m, ys_m, xe_m, ye_m]$, is created. The shaded areas in Fig. 4 represent these merged boxes. Using this merging algorithm, the proposed method is able to enlarge the candidate bounding box or make it smaller for its best fit to the existing object boundary.

When all the merged boxes from all the edge elements with overlap with the current candidate object are found, the search space for a more precise location is generated. In this step, the area in each merged box has warped into the determined image size used by CNN and is fed into this network for extracting its detection score and its object type.

The goal of the proposed optimization problem is to maximize the CNN detection score in each iteration of the search. Defining $\varphi(C, T)$ as a CNN detection score for an area inside the bounding box C to be from class T, the optimization problem in each iteration is represented in Equation 1.

$$\varphi(B, T_o) = arg \max_{C_e \in E} \varphi(C_e, T_i) \quad (1)$$

In this equation, C_e is a merged box obtained from merging the current candidate object with an edge element from the edge map E. The output of this optimization in each level is a modified candidate object's bounding box B, its class type T_o, and its detection score $\varphi(B, T_o)$, which are used as the inputs of the next iteration of the optimization algorithm.

The BFS algorithm searches for a merged box that improves the detection score of a recognized object. If the candidate objects score is positive, the search iterates for improving its detection score until there is no more edge element with overlap, or the detection score does not improve in several iterations. In this situation, the candidate object in each iteration is the optimized merged box of the previous one.

If the candidate object has a negative score, and at least one of its merged boxes with edge elements has positive score, the merged box with the highest score is chosen as a new candidate object and entered into the list of candidate objects. Otherwise, this candidate object is ignored as it probably does not have any recognizable object.

3. EXPERIMENTAL RESULTS

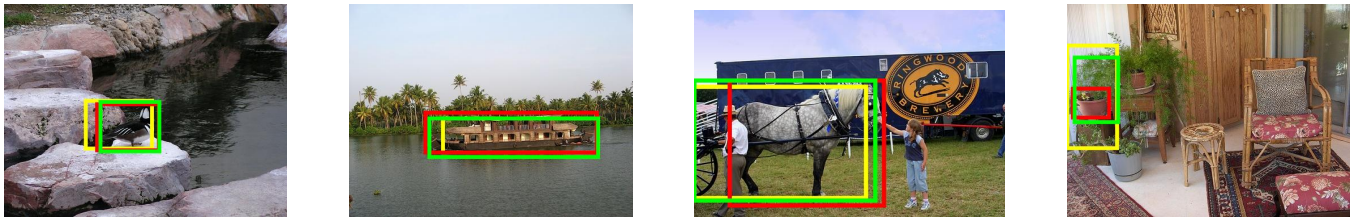
For evaluating the proposed method, we have selected the RCNN [8] with AlexNet Convolutional Neural Network [18] as a baseline model in all the experiments. It worth mentioning that the proposed method is independent from the underlying object recognition module. We have used Caffe [19] toolbox for training and implementing the proposed method. To find the edge elements, we have used the PCPG [17] package with the gradient threshold of 10, scanning interval of 8, and minimum edge length of 11 pixels.

We compared the performance of our proposed methods, such as RCNN with GET localization (GET_Loc), RCNN with trace localization (Trace_Loc), and RCNN with GET and trace localization (GT_Loc), with RCNN as a baseline method. To compare the proposed methods with the baseline model, we have tested all methods on the PASCAL VOC 2007 [20] and PASCAL VOC 2012 [21] datasets under the competition 4. The Mean Average Precision (mAP) of the proposed methods and the baseline for different classes of these datasets along with the average mAP for the entire datasets are represented in Table 1.

By comparing the overall mAP of the proposed method and the baseline model in both datasets, an improvement of approximately 3% is noticeable. The results show that the

Table 1: Comparison of mAP for different classes of (a) PASCAL VOC 2007 test set and (b) PASCAL VOC 2012 test set using the baseline RCNN model and different versions of the proposed model.

(a) 2007	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV	mAP
RCNN	49.8	**61.7**	32.8	25.2	24.2	**53.1**	61.5	49.0	22.8	48.8	33.2	39.4	51.4	51.5	48.4	15.6	**50.2**	35.0	**49.5**	51.2	42.7
GET_Loc	49.5	60.9	37.7	31.0	30.3	51.2	61.4	**54.4**	27.8	**53.7**	32.6	46.1	57.5	58.4	48.5	**20.8**	48.2	34.1	47.9	51.6	45.2
Trace_Loc	50.3	61.3	**39.8**	31.6	30.8	51.9	61.9	48.6	28.9	47.6	**34.3**	**47.1**	**58.7**	59.6	48.5	20.6	49.3	**35.5**	49.0	**52.2**	45.4
GT_Loc	**50.4**	61.3	39.4	**31.8**	**32.0**	52.3	**62.0**	48.9	**29.2**	47.8	33.3	46.8	58.4	**59.6**	**48.6**	20.7	48.4	35.4	49.2	51.9	**45.4**
(b) 2012	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV	mAP
RCNN	56.4	49.3	31.4	15.4	19.4	43.3	46.1	52.4	13.6	31.9	23.8	48.7	41.1	51.8	44.0	12.8	42.9	20.4	33.7	34.4	35.6
GET_Loc	**59.2**	52.7	**35.5**	**18.8**	22.7	46.0	49.0	55.1	17.2	**38.1**	26.4	51.3	**44.5**	53.8	47.0	**14.9**	44.7	**23.3**	**38.3**	**39.1**	**38.9**
Trace_Loc	58.4	**53.3**	35.2	**18.8**	22.5	46.5	48.6	54.9	16.6	37.8	25.8	**51.9**	43.7	**54.5**	**47.3**	13.8	44.3	22.2	37.8	38.4	38.6
GT_Loc	58.8	52.8	35.0	18.7	**23.1**	**46.8**	**49.1**	**55.2**	**17.5**	37.8	**26.5**	51.4	44.4	54.1	47.1	14.7	**45.3**	23.1	**38.3**	**39.1**	**38.9**



Bird
$(S_b, S_i) = (-0.02, 1.16)$
$(I_b, I_i) = (0.68, 0.88)$

Boat
$(S_b, S_i) = (-0.03, 0.65)$
$(I_b, I_i) = (0.75, 0.76)$

Horse
$(S_b, S_i) = (-0.47, 0.84)$
$(I_b, I_i) = (0.65, 0.74)$

Plant
$(S_b, S_i) = (-0.36, 0.06)$
$(I_b, I_i) = (0.20, 0.35)$

Fig. 5: Samples of ground truth object (Red) along with output images from RCNN (Yellow) and GT_Loc (Green). Comparing the detection scores and IoU of RCNN (S_b) and GT_Loc (S_i) represents more precise localization of the proposed method.

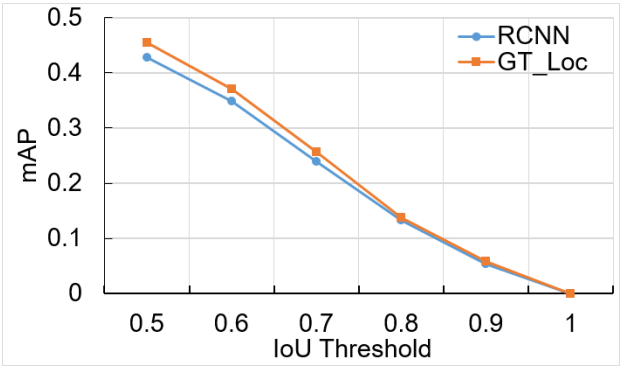


Fig. 6: Compare mAP of the proposed GT_Loc and the baseline model RCNN for different IoU

proposed method has improved the precision significantly for classes such as *'bird'*, *'boat'*, *'bottle'*, *'chair'*, *'dog'*, *'horse'*, *'motorbike'*, and *'plant'* where the edge information is precise in the images. This is concluded from the improved mAP of around 10% for these classes.

We also compared the mAP of the proposed and the baseline methods for different threshold values of IoU and the result is represented in Fig. 6.This diagram illustrates that the proposed method is more precise compared with the RCNN baseline method since in any of the overlap thresholds the proposed method has represented higher average precision.

Some examples of the objects recognized by GT_Loc are shown in Fig. 5 along with the outputs of the baseline RCNN model. These are some of the samples that RCNN has ignored due to their negative detection score, while the proposed method has improved their locations' precision which resulted in their positive detection scores and their recognition by the proposed method. Besides this much improvement, the proposed method takes about 5 seconds to process each object which is the major drawback of the proposed method that should be improved in the future.

4. CONCLUSION

The proposed method applies the BFS to the object localization and its search space, objective function and finishing conditions are specified. The search space is a set of edge elements whose overlaps with the current candidate object is greater than zero. The search iterates to optimize the detection score of CNN network as its objective function, and continues until there is no edge element with overlap or the improvement is stabilized. The proposed method has been tested on object recognition datasets and represents overall improvement compared with the baseline method of RCNN, while individual improvements for some classes are significant. As future work, there still exists the possibility for improving the object localization by using a combination of the image edge, color and texture information, and the learned features of the image.

5. REFERENCES

[1] Alexander Andreopoulos and John K Tsotsos, "50 years of object recognition: Directions forward," *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.

[2] Alexander Andreopoulos and John K Tsotsos, "A computational learning theory of active object recognition under uncertainty," *International journal of computer vision*, vol. 101, no. 1, pp. 95–142, 2013.

[3] Alexander Andreopoulos, Stephan Hasler, Heiko Wersing, Herbert Janssen, John K Tsotsos, and Edgar Korner, "Active 3d object localization using a humanoid robot," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 47–64, 2011.

[4] Ondrej Chum and Andrew Zisserman, "An exemplar model for learning object classes," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[5] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[6] Chaoyang Wang, Long Zhao, Shuang Liang, Liqing Zhang, Jinyuan Jia, and Yichen Wei, "Object proposal by multi-branch hierarchical segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3873–3881.

[7] Yao Xiao, Cewu Lu, Efstratios Tsougenis, Yongyi Lu, and Chi-Keung Tang, "Complexity-adaptive distance metric for object proposals generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 778–786.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[9] Ross Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[10] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[11] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, et al., "Deepid-net: Deformable deep convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2403–2412.

[12] Pedro Felzenszwalb, David McAllester, and Deva Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[13] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik, "Deformable part models are convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 437–446.

[14] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars, "Weakly supervised object detection with convex clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1081–1089.

[15] Xiaozhi Chen, Huimin Ma, Xiang Wang, and Zhichen Zhao, "Improving object proposals with multi-thresholding straddling expansion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2587–2595.

[16] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards, *Artificial intelligence: a modern approach*, vol. 2, Prentice hall Upper Saddle River, 2003.

[17] Qi-Gang Gao and AKC Wong, "Curve detection based on perceptual organization," *Pattern Recognition*, vol. 26, no. 7, pp. 1039–1046, 1993.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[20] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman, "The pascal visual object classes challenge 2007 (voc 2007) results (2007)," 2008.

[21] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman, "The pascal visual object classes challenge 2012 (voc2012)," 2012.