Object Detection for Crime Scene Evidence Analysis Using Deep Learning

Surajit Saikia[1,2(✉)], E. Fidalgo[1,2], Enrique Alegre[1,2], and Laura Fernández-Robles[2,3]

[1] Department of Electrical, Systems and Automation, University of León, León, Spain
{ssai,efidf,ealeg}@unileon.es
[2] INCIBE (Spanish National Cybersecurity Institute), León, Spain
l.fernandez@unileon.es
[3] Department of Mechanical, Informatics and Aerospace Engineering, University of León, León, Spain

Abstract. Object detection is the key module in most visual-based surveillance applications and security systems. In crime scene analysis, the images and videos play a significant role in providing visual documentation of a scene. It allows police officers to recreate a scene for later analysis by detecting objects related to a specific crime. However, due to the presence of a large volume of data, the task of detecting objects of interest is very tedious for law enforcement agencies. In this work, we present a Faster R-CNN (Region-based Convolutional Neural Network) based real-time system, which automatically detects objects which might be found in an indoor environment. To test the effectiveness of the proposed system, we applied it to a subset of ImageNet containing 12 object classes and Karina dataset. We achieved an average accuracy of 74.33%, and the mean time taken to detect objects per image was 0.12 s in Nvidia-TitanX GPU.

Keywords: Object detection · Convolutional neural network · Deep learning · Video surveillance · Crime scenes · Cyber-security

1 Introduction

The problem of detecting objects of interest in videos and images plays a key role in most security and surveillance systems. In the domain of forensic science, the digital images and videos play a significant role in determining fingerprints, identifying criminals and understanding crime scenes. For instance, given an object detected in an image, further analysis can be done to extract crucial information (i.e. relating objects of various crime scenes). However, due to the presence of a large amount of visual data, the creation of tools to manage or categorize this data have an exceptional importance. If we now imagine the task of a single person, trying to extract some type of intelligence from thousands of

S. Battiato et al. (Eds.): ICIAP 2017, Part II, LNCS 10485, pp. 14–24, 2017.
https://doi.org/10.1007/978-3-319-68548-9_2

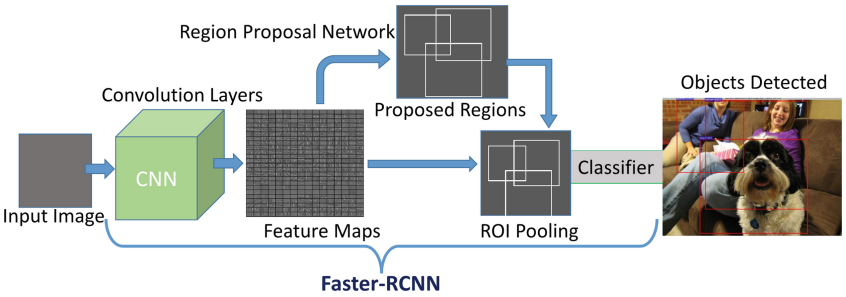

Fig. 1. Object detection pipeline using Faster R-CNN.

images or hours of video, i.e. a police officer reviewing digital evidence from a crime scene, the following question raises: *How is it possible to ease the task of detecting an object, that can be the evidence for a crime, inside this amount of visual information?* The answer could be an object detection system working in real-time.

Object detection is the task of detecting instances of objects belonging to a specific class. These systems are exploited in a wide range of applications in the field of AI (Artificial Intelligence), medical diagnosis [1,2], military [3] and crime prevention [4]. Furthermore, such systems can also be joined with other techniques to extract useful information in different types of cyber-crimes. Some examples might be Face detection and recognition [5] to detect and identify criminals on Internet videos, Video surveillance [6] to identify videos which can be a threat to society and nations, and Image Understanding [7] to recognize crime scenes based on the contents of images on the Internet.

In this work, we address the problem of analyzing the data gathered in a crime scene through the use of object detection. By detecting the objects present in the evidences found in a crime scene, it is possible to extract some intelligence or relations that can help a police officer, for example, to relate different crime scenes. We introduce an object detection method that uses the pre-trained VGG-16 [8] architecture of Faster R-CNN (Region-Based Convolutional Neural Network) [9], which was trained on MS-COCO [10] dataset. We selected 12 objects from ImageNet [11], which are most commonly found in a bedroom. Such objects are representative of the ones that might be found in an indoor environments. In Fig. 1 we briefly illustrate our object detection method. Given an input image, a Region Proposal Network (RPN) suggests regions based on the features generated by the last convolution layer. The proposed regions are then classified according to their detected class labels. Furthermore, this method can also be generalized for other objects and outdoor scenes. Our proposal aims at smoothing and reducing the labour of police while dealing with visual data (i.e. *a police officer can use this method to detect objects from one scene, and automatically relate them to other similar scenes efficiently*). To evaluate our proposal, we created a test-set and named it ImageNet-RoomObjects, which is a subset of

images from the ImageNet dataset. Furthermore, we also tested our method on the publicly available Karina dataset[1].

The rest of the paper is structured as follows. Section 2 briefly introduces the related works, Sect. 3 describes the method used to detect objects, Sect. 4 presents the experiments and results, and finally, Sect. 5 draws the main conclusions of this work.

2 Related Work

Recently in the literature, many works use deep convolutional neural networks (e.g. AlexNet [12], GoogleNet [13], VGG-Net [8]) for detecting and locating objects with class specific bounding boxes. Typically a CNN consist of multiple convolution layers, followed by ReLU (Rectified Linear Units), pooling layers and fully connected layers. The activations which are generated by the last layers of a CNN can be used as a descriptor for object detection and classification. Razavian *et al.* [14] employed the activations generated by the fully connected layers as region descriptors. Babenko *et al.* [15] demonstrated that such activations can be even used for image retrieval task, and they named such descriptors as neural codes. Later, they established that such descriptors performs competitively even if a CNN is trained for unrelated classification task i.e. a CNN trained with ImageNet [11] dataset can be generalized to detect objects in MS-COCO [10] dataset.

The deep learning algorithms have improved the image classification and object detection tasks in manifolds as compared to SIFT or other variants [16]. The algorithm proposed by Lee *et al.* [17] learns high-level features i.e. object parts from natural scenes and unlabeled images. Simonyan and Zisserman *et al.* [8] investigated the depth of such algorithms on its accuracy, and achieved state-of-the-art results.

Girshick *et al.* [18] presented R-CNN (Region-based Convolutional Neural Network), which proposes regions before feeding into a CNN for classification. The network is a version of AlexNet, which has been trained using Pascal VOC Detection data [19]. The network contains a three-stage pipeline, thus making the training process slow. Since then, in terms of accuracy and speed, great improvements have been achieved. He *et al.* [20] proposed SPP-net, which is based on Spatial Pyramid pooling. The network improves the detection and classification time by pooling region features instead of passing each region into the CNN. Later, Girshik [21] proposed Fast R-CNN, which is similar to SPP-net, but replaced SVM classifiers with neural networks. Ren *et al.* [9] introduced Faster R-CNN, a faster version of Fast R-CNN, which replaces the previous region proposal method with RPN (Region proposal Network), which simultaneously predicts object bounds and scores.

[1] http://pitia.unileon.es/varp/node/373.

3 Methodology

In this section, we introduce a system for object detection which is based on the Faster R-CNN algorithm. At first, we briefly describe the algorithm, and then we present the architecture details of VGG-16 network and the Region Proposal Network (RPN).

3.1 Background of Faster-RCNN

Region-based Convolutional Neural Network (RCNN) is an object detection method based on visual information of images. The network first computes the region proposal (i.e. possible locations of objects), and then it feeds the proposed regions into the CNN for classification. Nevertheless, this algorithm has important drawbacks due to its three-stage pipeline, which makes the training process expensive from space and computation point of view. For each object proposal, the network does a CNN pass without sharing the computations, thus making the network slow. As an illustration, if there are 1000 proposals, then we have to do 1000 CNN passes. In order to speed up the method, a faster version of R-CNN algorithm known as Fast R-CNN [21] was introduced. During CNN passes, this algorithm shares the computations when there are overlaps between the proposals, resulting in faster detection. Since the algorithm processes images by resizing them into a fixed shape, the detection time is approximately same for all the images. It takes approximately 2 s to detect objects including the time taken to propose regions.

Faster-RCNN. The Faster R-CNN [9] algorithm has brought the task of object detection to real-time, which takes approximately 12 ms for detecting objects in a RGB image, including the time cost in region proposal. The algorithm has replaced the previous region proposal method with Region Proposal Network (RPN), which is further merged with Fast R-CNN, making the network a single unified module.

3.2 VGG-16 Network

We use the 16 layered VGG [8] network, which comprises of 13 convolution layers and 3 fully connected layers. The network is unified with the RPN to serve the purpose of object detection. The Fig. 2 presents the architecture of the network excluding the RPN and RoI (Region of Interest) pooling layer, and the values represents the dimension of response maps in each convolution layer, i.e. in the first convolution layer $224 \times 224 \times 64$ represent 64 response maps of size 224×224.

Region Proposal Network. A RPN contains a sliding window of spatial size $n \times n$ (we use $n = 3$), which is applied on the feature maps generated by the last convolution layer to obtain an intermediate layer in 512 dimension. Then,

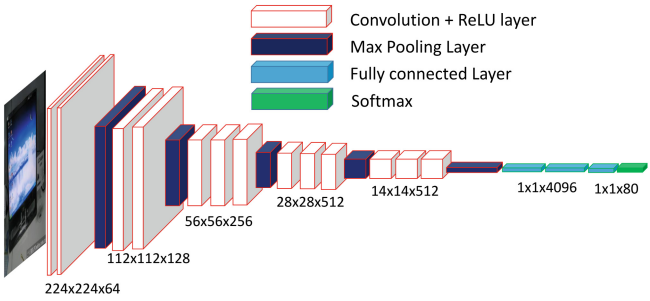

Fig. 2. Architecture of VGG net.

the intermediate layer feeds into a *box classification layer* and a *box regression layer*. There are k-anchor boxes with respect to each position of the sliding window, where k denotes the number of maximum possible proposals. The *box classification* layer determines whether the k anchor boxes contains object or not, and generates 2k scores (object/not object for each k). The *box regression layer* gives 4 coordinates with respect to each of the anchor boxes ($4k$ for k anchor boxes).

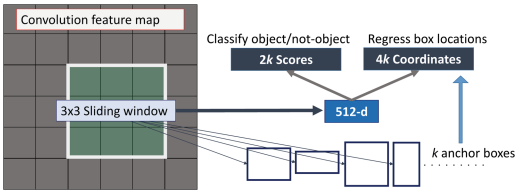

Fig. 3. Region proposal network (RPN).

The RPN is combined with a Convolutional Neural Network as a single module, which proposes regions within the network itself. It takes feature maps generated by the last convolution layer as an input and generates rectangular regions (object proposals) along with objectiveness scores. The RPN determines whether a region generated by a sliding window is an object or not, and if it is an object then the network does bounding box regression. Figure 3 shows the architecture of the RPN.

3.3 Object Detection

Once the complete system has been described, in Fig. 4 we present an example illustrating intermediate outputs of the system when we try to detect the objects. At first, an image (Fig. 4(a)) is given as input to the algorithm. The convolution layers generate activations, and the last layer activations are given as an input

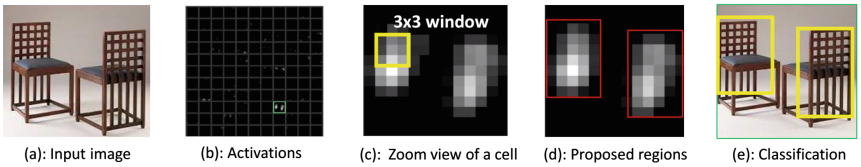

Fig. 4. Stages of object detection process.

to the RPN for region proposal. Figure 4(b) shows an example of the activation maps generated by the last convolution layer, and Fig. 4(c) presents how a cell in an activation maps looks like when it is zoomed. Next, the RPN slides a window of size 3×3 in each of the cell and fits the anchor boxes (Fig. 4(d)). Then, we classify the regions corresponding to each of the anchor boxes, and we obtain their class labels along with the bounding boxes (Fig. 4(e)). In this way, we can input images to the system to detect specific objects.

4 Experiments and Results

In this section, we describe the datasets, the experimental setup and the results achieved.

4.1 Datasets

Our network is pre-trained with the MS-COCO dataset. This is an image segmentation, recognition and captioning dataset by Microsoft corporation, which is a collection of more than 300,000 images and 80 object categories with multiple objects per image. Then, we tested the method in two different test-sets: a subset of images containing 12 indoor objects extracted from ImageNet, which we called it *ImageNet-RoomObjects*, and the *Karina dataset* [22]. We briefly describe the test-sets along with the datasets.

Test-Set: ImageNet-RoomObjects. This is a collection of 1345 images with 12 object categories, that are commonly found in an indoor environment, i.e. bedroom. We randomly selected images from the ImageNet, which is a huge dataset with a collection of 14,197,122 images with more than 1000 object classes, and each object class in this dataset contains thousands of images.

Karina Dataset [22]**.** This is a video dataset that was created to evaluate object recognition in environments which are similar to those that might appear in child pornography. The dataset contains 16 videos of 3 min which are filmed in 7 different rooms, and it contains 40 different categories, which represents to some of the objects that can be found most commonly in an indoor environment i.e. a bedroom.

Fig. 5. Examples of object detection results on ImageNet-IndoorObjects. Red squares overlaid on input images mark the bounding boxes of the detected objects. (Color figure online)

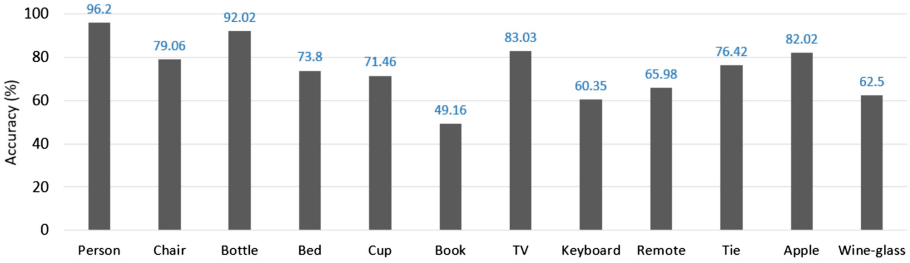

Fig. 6. Detection accuracy (in percentage) of each class.

4.2 Experimental Setup

We have used the pre-trained VGG-16 network architecture of Faster R-CNN to detect objects in our created test-set and the Karina dataset. The network was trained based on the following parameters: base learning rate: 0.001, learning policy: *step*, gamma: 0.1, momentum: 0.9, weight decay: 0.0005 and iterations 490000. All the experiments were carried out using the *Caffe* [23] deep learning framework in *Nvidia Titan X GPU* and in an *Intel Xeon* machine with 128 GB RAM.

4.3 Detection Accuracy

We present the detection accuracy (in percentage) of each class in the ImageNet-RoomObjects test-set, which is the percentage of true positives in each class of the test-set. Figure 6 shows the detection accuracies for all the 12 considered classes, and we obtained an average accuracy of 74.33%. We also present the mean of confidence scores (Fig. 7) for each class. Figure 5 shows the samples of object detection in each of the class category.

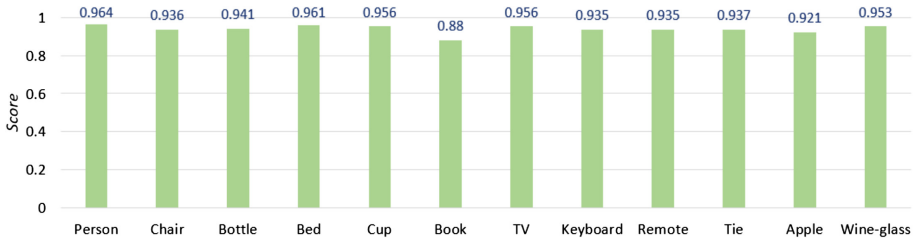

Fig. 7. Mean confidence scores per class.

Table 1. Detection time (in seconds) for each object class along with the number of images per class.

Class	Person	Chair	Bottle	Bed	Cup	Book	TV	Keyboard	Remote	Tie	Apple	Wine-glass
Time	31.35	26.8	37.02	35.6	21.65	21.65	27.22	31.62	24.74	28.27	27.94	34.53
Total images	104	100	110	129	121	120	130	110	105	114	106	105

4.4 Detection Time

This is the amount of time taken to detect specific objects in an image. Table 1 shows the time taken to detect objects specific to each of the class. For example, there are 104 images of person in the test-set, and the time needed to detect a person in those images was 31.35 s. The system takes 356.35 s in a GPU platform to detect objects in all the 1345 images, which also includes the time to propose regions. We also observe that, it takes an average of 2 s to detect objects in a single image in a CPU environment, and 0.12 s in a GPU environment, saving approximately 90% of the needed time.

4.5 Experiments on the Karina Dataset

To test the system, we have created a test-set from the Karina dataset by extracting image frames (size 640×480) from the videos. Out of 40 object categories we select 6 classes i.e. *bed, book, toy car, teddy, person and cup*. In Table 2 we present the detection accuracy for each class along with the total number of images present in each category. Figure 8 shows some samples, where we were able to detect some indoor objects like *(a) cup, (b) bed, (c) book, (d) toy car, (e) remote, (f) doll* and *(g) teddy bear*. Since the resolution of the images is low, we performed only preliminary test. In the future, we would like to apply *image super-resolution* techniques [24] to enhance the resolution before detection.

Table 2. Detection accuracy for each object class in the Karina dataset.

Class	Person	Book	Toy car	Teddy bear	Cup	Bed
Accuracy	81.3%	28.42%	5.4%	18.2%	45.61%	20.23%
Total images	100	120	105	80	109	91

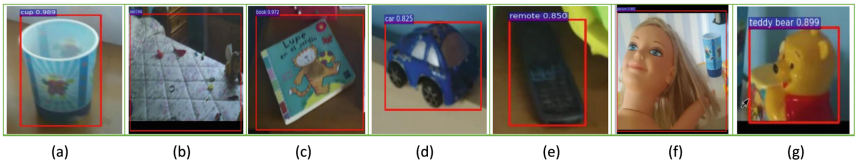

Fig. 8. Examples of object detection results on the Karina dataset. Red squares overlaid on input images mark the bounding boxes of the detected objects. (Color figure online)

4.6 Discussion

We have evaluated our object detection system in ImageNet-RoomObjects and the Karina Dataset, and we made several observations and conclusions based on our experiments.

First of all, our object detection system is pre-trained on the MS-COCO dataset. While testing on the ImageNet-RoomObjects, we observed that the system was able to detect objects with an average accuracy of 74.33%, where the highest accuracy was 96.2% obtained with the class *person*, and the lowest was 49.16% yielded with *books*. However, this performance can be further improved by fine tuning the architecture using datasets with respect to each class category, and we will address it in our future work.

We have also evaluated the performance of the system in the Karina dataset. But, due to the presence of low resolution images, the average accuracy obtained was 33.19%. In real world scenarios, we can not expect all the images to be of good resolution, and it is a difficult task for the forensic department to recreate crime scenes using such images. In the future, we will handle this issue by applying image-super resolution techniques to enhance the image quality, which will ease the task for police officers to detect objects effectively even in low quality images.

The average detection time per image was 0.12 s in Nvidia Titan X GPU, which makes the system suitable to be used as a real-time application. Furthermore, the system might be an application for a forensic department, which can help police officers to automatically detect objects of interest from large scale datasets in real-time.

5 Conclusion

In this work, we presented a real-time system which can detect objects related to indoor environments (i.e. bedroom). We have used the state-of-the-art VGG-16 network architecture of the Faster R-CNN algorithm, which can compute region proposal within the network itself. Due to this property, the algorithm can be widely used to develop real-time object detection applications. To evaluate the system, we have created a test-set "*ImageNet-RoomObjects*" comprising of images commonly found in an indoor environment, and we achieved state-of-the-art accuracy. The system has also been tested on the Karina dataset, but

we have achieved poor accuracy due to the low quality of the images. In future works, we will address this issue by applying image super-resolution techniques, and we will train a new model containing a large number of categories based on the additional object types that the police might found interesting during their crime scene research. Finally, the method can be used as a surveillance application to detect objects of interest in videos and images in real time for analysing various crime scenes.

Acknowledgement. This research was funded by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under addendum 22. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

References

1. Vaidehi, K., Subashini, T.: Automatic classification and retrieval of mammographic tissue density using texture features. In: 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), pp. 1–6. IEEE (2015)
2. Nosato, H., Sakanashi, H., Takahashi, E., Murakawa, M.: Method of retrieving multi-scale objects from optical colonoscopy images based on image-recognition techniques. In: 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 1–4. IEEE (2015)
3. Li, J., Ye, D.H., Chung, T., Kolsch, M., Wachs, J., Bouman, C.: Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs). In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4992–4997. IEEE (2016)
4. Rao, R.S., Ali, S.T.: A computer vision technique to detect phishing attacks. In: 2015 Fifth International Conference on Communication Systems and Network Technologies (CSNT), pp. 596–601. IEEE (2015)
5. Herrmann, C., Beyerer, J.: Face retrieval on large-scale video data. In: 2015 12th Conference on Computer and Robot Vision (CRV), pp. 192–199. IEEE (2015)
6. Sidhu, R.S., Sharad, M.: Smart surveillance system for detecting interpersonal crime. In: 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 2003–2007. IEEE (2016)
7. Vallet, A., Sakamoto, H.: Convolutional recurrent neural networks for better image understanding. In: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–7. IEEE (2016)
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
9. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
10. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). doi:10.1007/978-3-319-10602-1_48
11. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)

12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
14. Razavian, A.S., Sullivan, J., Carlsson, S., Maki, A.: Visual instance retrieval with deep convolutional networks. arXiv preprint arXiv:1412.6574 (2014)
15. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 584–599. Springer, Cham (2014). doi:10.1007/978-3-319-10590-1_38
16. Fidalgo, E., Alegre, E., González-Castro, V., Fernández-Robles, L.: Compass radius estimation for improved image classification using edge-sift. Neurocomputing **197**, 119–135 (2016)
17. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 609–616. ACM (2009)
18. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
19. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
20. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 346–361. Springer, Cham (2014). doi:10.1007/978-3-319-10578-9_23
21. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
22. Fernández-Robles, L., Castejón-Limas, M., Alfonso-Cendón, J., Alegre, E.: Evaluation of clustering configurations for object retrieval using SIFT features. In: Ayuso Muñoz, J.L., Yagüe Blanco, J.L., Capuz-Rizo, S.F. (eds.) Project Management and Engineering. LNMIE, pp. 279–291. Springer, Cham (2015). doi:10.1007/978-3-319-12754-5_21
23. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
24. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Cham (2014). doi:10.1007/978-3-319-10593-2_13