Deep Learning for Fast and Accurate Fashion Item Detection

Evgeny Smirnov
Kuznech Inc.
Evgeny.Smirnov
@kuznech.com

Anton Kulinkin
Kuznech Inc.
Anton.Kulinkin
@kuznech.com

Karina Ivanova
Kuznech Inc.
Karina.Ivanova
@kuznech.com

Michael Pogrebnyak
Kuznech Inc.
Michael.Pogrebnyak
@kuznech.com

ABSTRACT

This paper proposes fast and accurate fashion item detection model, based on deep neural networks. The model combines MultiBox and Fast R-CNN detection architectures and improves them with several modifications, intended to make object detection system faster while keep detection quality at the same or better level. The model was tested on Kuznech-Fashion-156 and Kuznech-Fashion-205 fashion item detection datasets and gave good detection results while being 10 times faster than baseline model. Image processing time for one image is 310 ms. Obtained results make it possible to use this model as a part of Kuznech Mobile Recognition system for the task of fashion item detection and recognition, followed by visual similarity search.

CCS Concepts

•Computing methodologies → Object detection; Neural networks; Object recognition;

Keywords

Object Detection; Deep Learning; Convolutional Neural Networks

1. INTRODUCTION

Growth of online commerce, specifically m-commerce, and demand on fashion-related products are the triggers that stimulate scientific research in the field of computer vision for fashion and luxury market segments. One of the most important and challenging tasks in this area is detection [8, 21, 11] and recognition [1] of apparel and accessories in real-world images, followed by search for similar items [11, 14, 7, 21] in online shops. This task is well known but for a long time there were no good solutions, which are simultaneously accurate and fast. In this paper we present **Kuznech Mobile Recognition** system, a model, which can accurately detect all fashion items in photo, classify each of them and find visually similar items in large database, and all that in a very short period of time (under 0.5 second). Out model handles more than two hundred categories of fashion items of three basic classes: clothing, footwear and bags. The whole system is composed of several modules, most of them are beyond the scope of this paper. Here we will describe only fashion item detection and classification module. The demo of our system is available on the web[1].

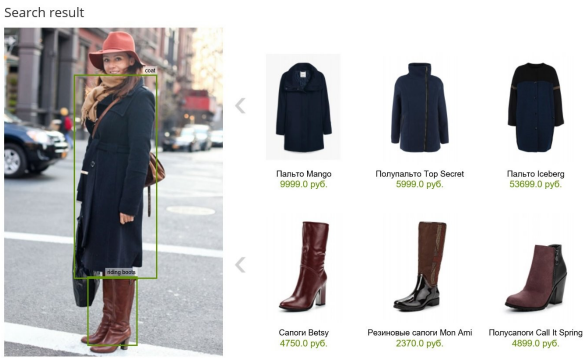

Figure 1: Example of fashion item detection and similarity search, performed by Kuznech Mobile Recognition system

2. RELATED WORK

Usually the most challenging part of fashion item detection, recognition and similarity search systems is object detection, when the system must find, localize and classify all objects of some set of categories, which are presented on image. Until recently methods used for this task were not accurate enough, especially in real-world conditions and for objects with very diverse visual appearance like clothing. One way to fight this problem is to use recently proposed object

[1]http://mobilerecognition.kuznech.com/

ML4Fashion '16 August 13, 2016, San Francisco, CA, USA

DOI: 10.475/123_4

detection methods based on Convolutional Neural Networks [13]. In the last couple of years **Convolutional Neural Networks** proved to be the best method for visual recognition problems, even outperforming humans in some image classification challenges [9]. There are several ways how Convolutional Neural Networks could help in object detection. One of them is **R-CNN** [6], model for object detection, which consists of bottom-up region proposal phase and region classification phase. This model was used for fashion item detection in [8]. Similar models also were used for bag detection [4, 2]. The problem of this model is that it takes too much time to detect objects. Later authors proposed better version of this model - **Fast R-CNN** [5]. It worked much faster than original R-CNN and could be used in real applications. We used this model as a baseline for our system and improved it in several ways. Our final model works **10 times faster** than Fast R-CNN, keeping similar level of detection accuracy. In the following parts of the paper we'll describe how we achieved this.

3. PROPOSED METHOD

In most papers, which use R-CNN or Fast R-CNN for object detection [6, 5, 7, 8, 2] method called **Selective Search** [20] is used for region proposals generation. It is good but slow method, it takes 2.6 seconds to generate all proposals needed for one image. This is why our first modification to the standard Fast R-CNN model was switching Selective Search to different method - modified version of **MultiBox** [19].

3.1 Modified MultiBox for proposal generation

MultiBox is a neural network that receives an input image and generates N region predictions - coordinates of bounding boxes and values of network's confidence that this particular region (bounding box) contains an object. MultiBox is trained with backpropagation as usual neural network. It differs from usual neural networks, used for classification tasks, in its "head" with loss functions: it has two separate branches, one for predicting region coordinates (it is trained with L2 loss function in original paper) and another for predicting confidence values (in original paper it is trained with Softmax). We discovered that if we change **L2 Loss function** (1) to **Smooth L1 Loss function** [5] (2,3) then network starts to generate better region proposals.

$$F_{match}(x, l) = \frac{1}{2} \sum_{i,j} x_{ij} \left\| l_i - g_j \right\|_2^2 \tag{1}$$

$$L_{loc}(t^u, v) = \sum_{i \in x,y,w,h} smooth_{L1}(t_i^u - v_i) \tag{2}$$

$$smooth_{L1}(x) = \begin{cases} 0,5x^2 & , \|x\| < 1 \\ \|x\| - 0,5 & , \|x\| \geqslant 1 \end{cases} \tag{3}$$

Also we switched standard Softmax loss to its modified version with **hard bootstrapping** from [16] to ensure better tolerance to noisy labels. Besides, to increase model training speed we used less neurons in the bottleneck layer of MultiBox network and initialized neural network's weights with weights, taken from **GoogLeNet** [18], pre-trained on

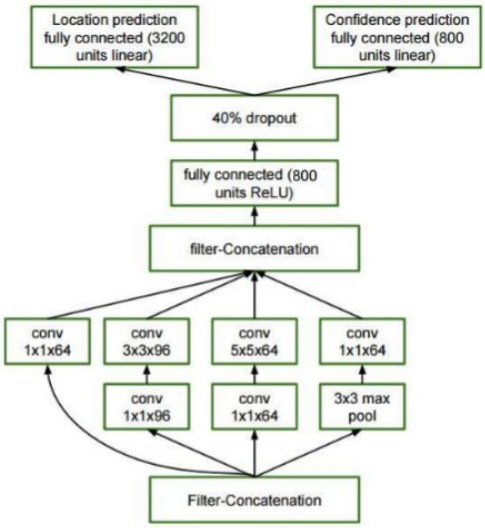

Figure 2: Architecture of MultiBox network's output layers

ImageNet dataset, and then fine-tune the network on our dataset (in original paper MultiBox was trained from scratch).

As in original paper, we used adjustable threshold for confidence values of MultiBox to filter object proposals that obviously do not contain any objects. To make object predictions more accurate, especially for very small objects, we used MultiBox in spatial pyramid fashion: instead of just using one full image as input we used 14 of its different crops in several scales. This slows detection but ensures very high accuracy.

3.2 Region classification

As a result of MultiBox a certain amount of region proposals is generated. This is an equivalent of the outputs of Selective Search method in original Fast R-CNN model, so for each proposal box, we need to define whether it contains an object, and if yes, of which class. For this purpose convolutional neural networks (CNN) are used. In older R-CNN model [6] each resulting region proposal box was cut from input image, brought to the same size and put through a neural network, which outputs a vector of N+1 values corresponding to the probability of the box's belonging to each of N classes or the "background" class (non-presence of any object). In case if it is required to evaluate a big number of boxes (and for Selective Search it is more than thousand boxes), this resulted in considerable time consumption. The main difference of Fast R-CNN model is that the neural network does not treat each box prediction individually, but the image itself as a whole. Passing through a sequence of convolutional layers to the upper levels of the network, the image is converted into a set of feature maps. In this case, box predictions are cut not from input image, but from feature maps, and then one by one go through only the uppermost layers of the network. Thus, a considerable acceleration of network performance is obtained, since instead of thousand image passes through the full network it is required to make only one pass. Furthermore, by using MultiBox for generating box predictions, it is possible to reduce the number

Figure 3: Non Maximum Suppression applied to the detection results

of used boxes from thousands to hundreds or tens by beforehand setting the confidence threshold value which will significantly raise system performance. Unlike in the Fast R-CNN version, described in original paper [5], as a base network we used GoogLeNet [18], which is faster than network from Fast R-CNN paper and which, as in case with MultiBox, was trained on the task of ImageNet classification and later fine-tuned on our detection task. Also we placed **ROI Pooling layer** after 7th Inception block instead of last convolutional layer as in original paper. This way we slow down the network's work for a bit but make it more suitable for fine-tuning. Additionally, due to a relatively small size of a training set (100-200 images per class), we actively used various distortions (mirroring, image stretching by the width and height, image cropping, etc.) as a way of **data augmentation**.

3.3 Post-processing of detection results

We used a **Non Maximum Suppression** method (NMS) [6] to exclude unnecessary detection of one and the same object at the stage of results post-processing: among much intersecting fields classified as containing objects of the same class, we selected areas with the highest probability of containing an object of this class.

4. EXPERIMENTS

4.1 Dataset

To train a MultiBox network and region classifier network we collected and manually labelled with bounding boxes a dataset of 25,000 images, which contains various items of clothing, shoes and accessories - 156 classes in total. For testing we used another set of 6,000 images. We'll refer to this dataset as **Kuznech-Fashion-156**. Later we updated this dataset with more classes (205 classes in total) and more images (40,000 for training, 12,000 for tests) We'll refer to this updated dataset as **Kuznech-Fashion-205**. While the total number of images in these datasets is not very large compared with datasets like ImageNet, they present very diverse sets of training and testing data ranging from very high-quality advertisment pictures to low resolution images from mobile phones. Some images contain only one object of one single category, others have a lot of different object from different categories. Since there are only 100-200 images per class, and a lot of classes are similar, to get good results it is important to use data augmentation methods or neural networks pretrained on large datasets.

Table 1: Comparison of MultiBox + Fast R-CNN model with Selective Search + Fast R-CNN model

Model	Processing Time, sec
Selective Search + Fast R-CNN	3
MultiBox + Fast R-CNN	**0.3**

Table 2: Region proposal generation methods

Method	Processing Time, sec
Selective Search [20]	2.6
GOP [12]	0.6
Edge Boxes [23]	0.25
MultiBox	**0.04**
MultiBox (spatial pyramid)	**0.14**

4.2 Results

Evaluation of the system work quality was made according to two main guidelines: quality of object detection and amount of processing time. Our main goal was to create a model with good detection precision and recall, which could process one image in less than 0.5 second, so it could be used in real applications. As a result we achieved average precision of **81%** and recall of **85.5%** on dataset Kuznech-Fashion-156. To process one image it takes about **310 ms**. Comparison of the proposed detection model with original Fast R-CNN + Selective Search model is shown in Table 1. Comparison of our modified MultiBox method with other methods of region proposal generation is shown in Table 2. Later we trained our model on Kuznech-Fashion-205 dataset and achieved average precision of **81.25%** and recall of **78%**.

5. CONCLUSIONS

Using Fast R-CNN model combined with the modified MultiBox method for region proposal generation, we built a system, that quite reasonably and accurately detects objects of 156 and 205 classes in the images, and works fast enough to be used in real applications. The resulting working speed was several times higher than that of existing analogs (at the time when experiments were conducted). Thanks to pretrained classification network, which we used to initialize our models, modified loss functions in MultiBox and additional data augmentation techniques, we were able to successfully train the system on a relatively small dataset. This model is used as a part of **Kuznech Mobile Recognition** system, combined with algorithms for visual similarity search. However, at this point of time the system is still not perfect. To predict regions containing objects and classify them, we use two different networks, which requires replicated computation of low-level features. By combining these two networks into one, we could use same features on the step of prediction boxes and object classification. Thus, computational costs could be considerably reduced. Similar things were performed in [17, 15]. Also the system could be improved by using better neural network architectures [10, 22] and activation functions [3].

Figure 4: Example of detection and similarity search, performed by Kuznech Mobile Recognition system

6. REFERENCES

[1] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool. Apparel classification with style. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part IV*, ACCV'12, pages 321–335, Berlin, Heidelberg, 2013. Springer-Verlag.

[2] C. Cao, Y. Du, and H. Ai. Bag detection and retrieval in street shots. In *MultiMedia Modeling*, pages 780–792. Springer, 2016.

[3] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[4] Y. Du, H. Ai, and S. Lao. A two-stage approach for bag detection in pedestrian images. In *Computer Vision–ACCV 2014*, pages 507–521. Springer, 2014.

[5] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[7] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to buy it: Matching street clothing photos in online shops. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3343–3351, 2015.

[8] K. Hara, V. Jagadeesh, and R. Piramuthu. Fashion apparel detection: The role of deep convolutional neural network and pose-dependent priors. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016.

[11] Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel. Visual search at pinterest. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1889–1898. ACM, 2015.

[12] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *Computer Vision–ECCV 2014*, pages 725–739. Springer, 2014.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] K. Lin, H.-F. Yang, K.-H. Liu, J.-H. Hsiao, and C.-S. Chen. Rapid clothing retrieval via deep learning of binary codes and hierarchical search. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 499–502. ACM, 2015.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015.

[16] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[19] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441v2*, 2014.

[20] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[21] K. Yamaguchi, T. Okatani, K. Sudo, K. Murasaki, and Y. Taniguchi. Mix and match: Joint model for clothing and attribute recognition. In X. Xie, M. W. Jones, and G. K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 51.1–51.12. BMVA Press, September 2015.

[22] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[23] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.