Clothes Detection and Classification Using Convolutional Neural Networks

Jan Cychnerski*†, Adam Brzeski*†, Adrian Boguszewski*, Mateusz Marmołowski* and Marek Trojanowicz*
* CTA, Gdańsk, Poland, email: contact@cta.ai
† Gdańsk University of Technology
Faculty of Electronics, Telecommunications and Informatics
Gdańsk, Poland

***Abstract*—In this paper we describe development of a computer vision system for accurate detection and classification of clothes for e-commerce images. We present a set of experiments on well established architectures of convolutional neural networks, including Residual networks, SqueezeNet and Single Shot MultiBox Detector (SSD). The clothes detection network was trained and tested on DeepFashion dataset, which contains box annotations for locations of clothes. Classification task was evaluated on a set of images of dresses that were collected from online shops. Ground truth labels were inferred from shop items metadata for five different attributes, including color, pattern, sleeve, neckline and hemline, each consisting of several possible classes. Automatic gathering of labels resulted in an average of 83% rate of correct labels. In the experiments we evaluate the impact on classification accuracy of a set of potential improvements, including data augmentation by generating diverse backgrounds, increasing the size of the network and using ensembles. We analyse the accuracy improvements with respect to the processing efficiency. Finally, we present the achieved accuracy rates in the clothes detection task and outline the most successful network configurations for dresses classification.**

I. Introduction

Issues related to fashion and clothing are an important element of human culture. Nowadays, fashion is also an important part of the economy, including virtual economy on the Internet. Customers expect online stores to provide them with an easy way to find clothes that match their tastes. Therefore, there is a need for high quality clothing search engines. On the other hand, clothing suppliers are not sufficiently prepared to add their products to such search engines because it would require a very accurate, systematic and unified description of their products. Moreover, each store or search engine has a different set of categories and attributes, which is not compatible with others. In order to place a product in a search engine it is necessary to assign it to the appropriate category and apply correct attributes, as in figure 1.

In the past years, the problem was solved by manual labeling, and sometimes by constructing classifiers based on manually generated descriptors, such as [1]–[3]. Due to the fact that clothing in online stores is usually well photographed (studio-quality, solid white background), a promising technology for this purpose is deep learning, especially deep convolutional neural networks, which are proven to be highly successful in classifying images [4] and clothing recognition [5].

There are many generic algorithms that use deep neural networks, which allow recognizing similar elements in images as in the described problem. For example, the Faster R-CNN [6] or SSD [7] provide general detection and localization of objects on images. Moreover, ResNet [8], SqueezeNet [9], GoogLeNet [10] provide general image classification. There are also many other uses of deep neural networks, such as searching for similar clothes [11], [12] or people identification based on their clothing [13].

In this article, adaptation of existing deep learning methods to the problem is presented. The utilized databases and methods of data preparation are described. Finally, results of experiments proving effectiveness of these methods are shown.

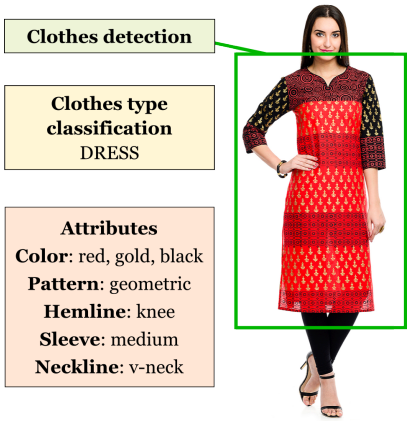


Fig. 1. Clothing detection, classification & attribute assignment

II. Related work

There are many distinct problems in clothing image analysis, including:

- detection — rough localization of clothing elements, usually by rectangle(s) [5], [12]
- segmentation — exact localization, often pixel-by-pixel [14]–[16]
- classification — assigning clothing to one of few classes, e.g. type of clothing (dress, skirt, shoe etc.) [5], [17], [18]

978–1–5090–6505–9/17/$31.00 © 2017 IEEE

- attribute assignment — assigning many of many attributes to the clothing at once, e.g. patterns, colors, styles, sizes etc. [5], [13], [19]–[22]
- describing — description of clothing in natural language [13]
- modeling — recognizing spatial structure of clothing [23]–[25]
- retrieval — searching for similar (in some aspect) clothes [1], [5], [11], [12], [19], [20], [22]

All of these problems are in interests of researchers since many years. Over time, methods used to solve them evaluated from hand-made complex mathematical models to most recent deep learning approaches. There are 3 main types of methods used:

- formula-based — based on arbitrary clothing mathematical models, created manually to resolve some problem in clothing image analysis [15], [24]
- traditional features learning — methods based on manually created simple features (like HOG LBP, SIFT, which are then processed by simple machine learning models: SVM, Bayesian, Random Forest etc. [3], [26]
- deep features learning — based on deep neural networks trained with big amount of varied data. This is the most recent and promising method. Using large image datasets (80k in [18], [27], 160k in [22], 245k in [21], 400k in [12], 800k in [5], over million in [13], [17], [20]) there have been achieved high accuracies in many problems, including detection and classification (78% [17], 60% [27]), attribute assignment (48%–76% [13], 35 attrib. over 80% and 93 attrib. 50–80% [22], 60–80% [20], 43–86% [21], 49–73% top 5 [5]), clothing retrieval (14% top 5 [5])

In this paper, the third method is evaluated using best known general deep learning models, modified to fit clothing detection, classification and attribute assignment.

III. Datasets

Clothes detection experiments were carried out on 290,000 images from Attribute Prediction subset of DeepFashion dataset [5]. Each image in the set is annotated with type of the clothing and a corresponding bounding box. We divided this set into train set containing 250,000 images and test set containing 40,000 images. Number of images for each clothing type is presented on secondary axis in figure 3.

Dresses classification experiments were evaluated on images crawled from online shops. We also collected text metadata of the products in order to infer labels for five different attributes of dresses including color, pattern, sleeve-type, neckline and hemline. We assumed color to be a multi-class attribute and the remaining attributes to be single-class. The lists of possible values for each attribute are presented in table I. Automatic inferring of labels, however, is a difficult task and may lead to inaccurate results. Table II presents the number of collected images with automatically inferred labels for each attribute as well as the estimated ratio of incorrect labels, measured on random samples of 200 images for each attribute. We also measured the ratio of bogus images in the dataset, which are hard to be automatically detected and excluded, though significantly worsening classification results. Wrong images, however, turned to be rare, except for the neckline attribute, where many of the collected images presented a rear view of the dress, which unfortunately prevents successful identification of the neckline type. In case of the color attribute images, apart from the automatically inferred labels we also performed significant manual assignment of labels, hence the lowest wrong labels rate.

TABLE I
DRESS ATTRIBUTES AND ACCORDING CLASSES USED IN THE CLASSIFICATION TASK

Attribute	Type	Classes
Color	Multi-class	black, blue, brown, gray, green, multicolor, orange, pink, red, violet, white, yellow
Pattern	Single-class	floral, geometric, graphic, plaid, polka-dot, solid, stripes
Sleeve type	Single-class	34-sleeve, long-sleeve, short-sleeve, sleeveless, strapless
Neckline	Single-class	boat-neck, collared, halter, high-neck, off-the-shoulder, one-shoulder, round, square-neck, sweetheart, turtleneck, v-neck
Hemline	Single-class	above-knee, high-low, knee, long, midi

TABLE II
STATISTICS OF THE DATASETS USED IN THE CLASSIFICATION TASK

Attribute	#Imgs	+/- wrong labels	+/- wrong images
Color	43 265	10%	<1%
Pattern	107 005	22%	<1%
Sleeve type	133 060	15%	1%
Neckline	178 011	23%	23%
Hemline	122 960	13%	<1%

IV. Methods

A. Clothes detection

Clothes images in general not always present a single piece of clothing, but may include many different clothes of different or the same kind, as more than one person can be present in the image. Such images cannot be directly classified in an accurate way. For this reason, it is necessary to detect pieces of clothing in the image before they can be individually classified.

There are many state-of-the-art solutions dedicated to general object detection on images, such as: Faster R-CNN [6], PVANet [28], R-FCN [29] and SSD [7]. Basing on comparison of their performance on PASCAL VOC2007 and VOC2012 datasets [30], we decided to choose SSD300 for our clothes detection problem, due to very good mAP (77.2%) reported by the authors, and high detection speed (46 FPS on Titan X Pascal GPU) [7].

We trained SSD from scratch using our train set. Most hyperparameters were set to default values for SSD. Initial

learning rate was 0.001 and was decreasing ten times every time network stopped to learn. The test interval was set to 1 epoch.

After 135 epochs of training, model accuracy plateaued at mAP of 42%. In figure 3 we show clothes detection results, evaluated as the accuracy of identifying multiple clothes in the image, without evaluating precision of the boxes. Some classes e.g. anorak, jeggings or halter resulted in a very low F-score, but this is due to the fact, that they are subclasses of other clothes, which negatively influences detection scores. Still, general types of clothes such as dress, jeans or skirt are detected with very high accuracy and F-score exceeding 0.9.

Fig. 2. Sample clothes detection results. From left: Dress - 0.965, Jacket - 0.863, Skirt - 0.998

B. Dress classification

In the initial experiments we trained SqueezeNet 1.1 [9] networks for each attribute to detect the expected classes. We split the images sets to train, val and test sets with ratios of 0.8/0.1/0.1. We then balanced the datasets by duplicating images within the smaller classes. The training of single-class attributes was conducted with Softmax loss, while for color attribute we used Euclidean loss. We performed hyper-parameter search for determining optimal training parameters for each attribute. Finally, we selected models with we lowest loss computed on the validation sets.

The results of the initial training were presented in table III. For each of the attributes we measured classification accuracy on the test set, except for the color attribute, which is of multi-class type. We evaluate the result of color classification as an F-score, computed as an averaged harmonic mean of sensitivity and specificity for each of the classes. Except the results acquired on the original test, that is affected by the mentioned earlier wrong labels ratios, we also present approximate results on small test sets with manually corrected labels, each consisting of 200 samples.

TABLE III
INITIAL CLASSIFICATION RESULTS USING SQUEEZENET NETWORK

Attribute	Test set result	Test on correct labels
Color	62.8%	64.5%
Pattern	66.9%	60.5%
Sleeve type	78.7%	88.5%
Neckline	61.8%	74.0%
Hemline	73.4%	83,0%

V. BACKGROUND INDEPENDENCE

Like most of deep learning problems, also in this topic it is crucial to have big and representative amount of data to successfully train the neural network. The data that we have consists mostly of studio images with white background, thus making it difficult to train the NN to process real-world images [31].

We have tested if substitution of white studio background with other backgrounds — which operation we call *background augmentation* — allows the NN to correctly recognize colors of more real-world-like images of clothing (the same problem as recognizing *Color* attribute described in table I).

A. Method

The algorithm of *background augmentation* is as follows (see: algorithm 1). At first, we check if the photo has white background with algorithm 2. If so, we remove white background from the image with algorithm 3. Then, we generate random background image with alorithm 4. At the end, we blend the image with background and apply slight simple transforms. Simple transforms consist of: random change of hue, brightness and contrast, and additive RGB noise. The whole algorithm is shown in figure 5. Sample results of the algorithm are shown in figure 6.

Algorithm 1 Background augmentation
1: **if** image has no white background **then**
2: **return** error
3: **end if**
4: roughly detect person or dress position
5: center and resize it to match 512×512 image
6: detect white background
7: get random background image
8: blend the image with background
9: apply slight simple transforms

Algorithm 2 White background identification
1: blur image with 11×11 pixels kernel
2: take into account only regions 10 pixels near the image border
3: calculate 3D (RGB) histogram
4: check if the last histogram bucket has the most pixels

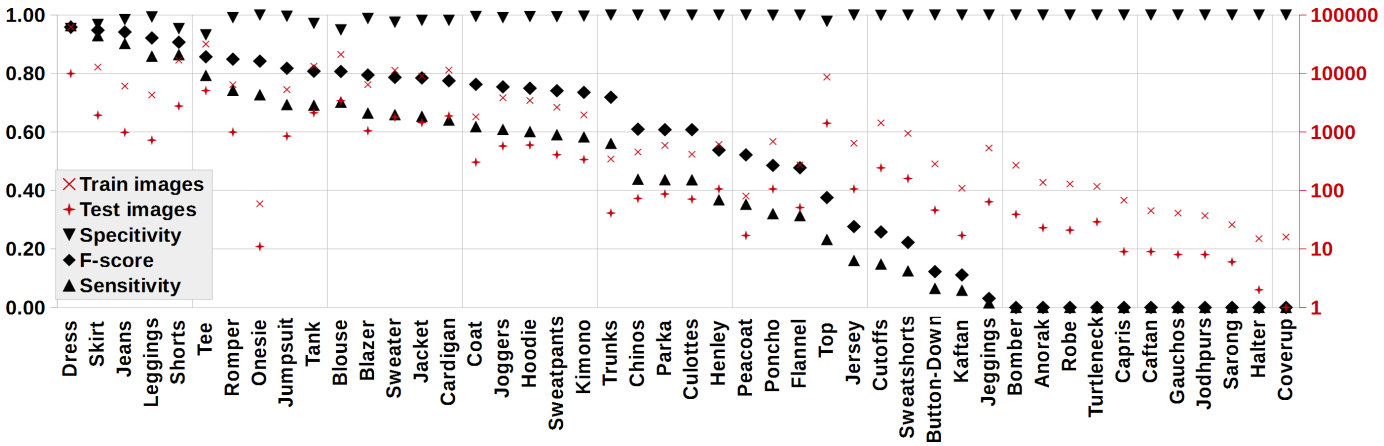


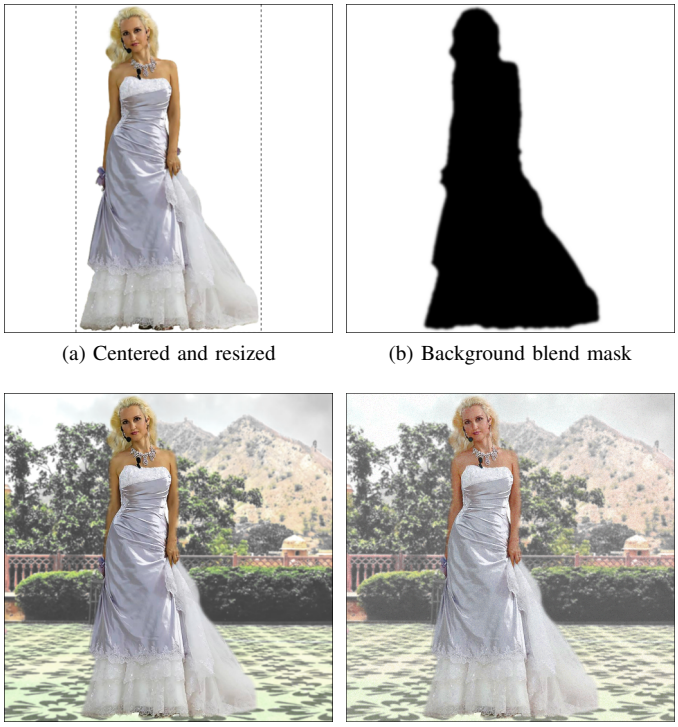
Fig. 3. Clothes classification test results (primary axis) and dataset sizes (secondary axis)

Algorithm 3 White background removal

1: *mask* ← pixels from *image* brighter than 250
2: dilate *mask* by ellipse 3×3
3: erode *mask* by ellipse 3×3
4: dilate *mask* by ellipse 17×17
5: erode *mask* by ellipse 23×23
6: gaussian blur *mask* with 11×11 kernel
7: alpha channel of image ← inverted *mask*

Algorithm 4 Background generation

1: randomly choose one of the backgrounds
2: randomly choose square subimage of it
3: resize to 512×512
4: apply strong simple transforms



(a) Centered and resized (b) Background blend mask

(c) Person and background blended (d) Simple modifications applied

Fig. 5. Background augmentation algorithm steps

B. Data preparation

For tests we used *Color* dataset described in table II, which consisted of 43265 dress images from open sources on the Internet, with each image manually labeled with color names. As dresses are often multi-colored, many labels could be assigned to one image. For best results of CNN training, we balanced the set by duplicating images, so that each color is present on approximately 5000 images. After that, the dataset contained 65,000 images total, which is 5000 images for each of 13 colors. Dataset is described in table IV.

For background augmentation, we collected a set of 200 photo background images.

C. Test configurations

Background augmentation was tested using SqueezeNet 1.1 implemented in Caffe framework. Input images were scaled down to 227×227 RGB images. At each test, random values of metaparameters were chosen, and neural net was trained for 1 hour on a Titan X Pascal GPU. Training was tuned to use no less than 90% of GPU's processing power.

We tested four configurations of background augmentations, described in table V. All configurations shared original image sets (described in table IV). Those original image sets were modified with augmentation algorithms. At each test, neural net was trained on *Train* set and tested on two validation sets (*Val A* and *Val B*). *Train* and *Val A* sets were created using configuration-specific augmentation. *Val B* set was a reference dataset, shared by all configurations. Sample images of test configurations are presented in figure 6.

For each configuration, 60 tests were performed, from which the one returning highest F-score on *Val B* set was chosen as a result. Then, these nets were tested on *Test A* and *Test B* sets to get more reliable results.

✓ strapless	✓ 3/4 sleeve	✓ 3/4 sleeve	✓ sleeveless	✓ sleeveless
✓ long	✓ long	✓ midi	✓ midi	✗ high-low
✓ sweetheart	✓ one-shoulder	✓ v-neck	✗ collared	✗ high-neck
✓ solid	✓ solid	✗ polka-dot	✓ graphic	✓ graphic
✓ white	✓ black	✓/✗ red, white	✓ blue, multi, red	✓ multicolor, pink

✓ short sleeve	✓ sleeveless	✓ strapless	✗ strapless	✓ long sleeve
✓ above knee	✓ above knee	✓ long	✓ knee	✓ midi
✓ round	✓ square-neck	✓ off-the-shoulder	✓ square-neck	✓ collared
✓ solid	✓ solid	✗ floral	✗ stripes	✓ solid
✓ black	✓ red	✓ gray, white	✓ red, white	✓ violet

Fig. 4. Sample dress classification results acquired using Resnet-50 network

TABLE IV
BACKGROUND AUGMENTATION DATA SET

Color	Train	Val	Test	Total	Aug. ratio
black	4000	500	500	5000	1.00
white	4000	500	500	5000	1.00
red	4000	500	358	4858	1.03
blue	4000	423	394	4817	1.04
gray	3542	467	381	4390	1.14
pink	3513	262	256	4031	1.24
beige	2765	346	352	3463	1.44
multicolor	2662	451	340	3453	1.45
violet	2005	292	218	2515	1.99
green	1875	234	168	2277	2.20
orange	886	151	127	1164	4.30
brown	908	129	120	1157	4.32
yellow	969	93	78	1140	4.39
TOTAL	**35125**	**4348**	**3792**	**43265**	$\Sigma = 65000$

TABLE V
BACKGROUND AUGMENTATION TEST CONFIGURATIONS

Images:	*Train*	*Val*		*Test*	
Configuration	**Train**	**Val A**	**Val B**	**Test B**	**Test A**
Base	none				
Noise	simple transforms				none
Background	background substitution				
Background + noise	background substitution + simple transforms				

D. Results

Intuitively, for more disturbed by augmentation images, neural net training results should be lower than for less disturbed images, as the recognition task is more difficult. On the other hand, their generalization performance should be higher.

Our tests gave results similar to these intuitions. As can be seen in figures 7 and 8, efficiency on *Test A* set (with white backgrounds) was much worse for configurations enabling background augmentation. At the same time, these configurations gave better results on *Test B* set, which was created with maximal augmentation possibilities. This proves that our nets trained on more *disturbed* data have more generalization

(a) Base configuration

(b) Noise configuration

(c) Background configuration

(d) Background+noise configuration

Fig. 6. Sample images for test configurations

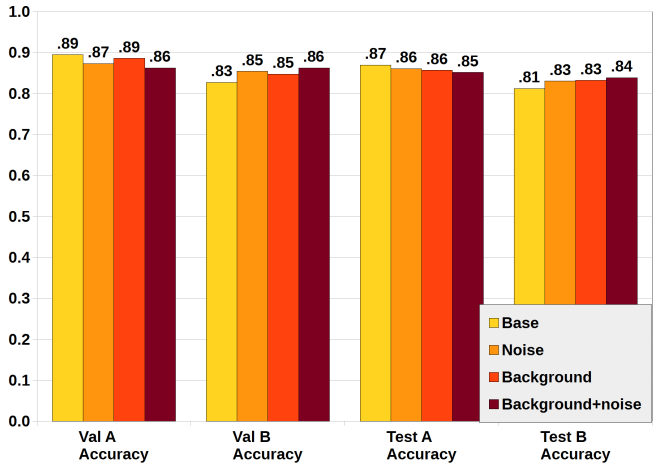


Fig. 7. Accuracy on *Val* and *Test* sets

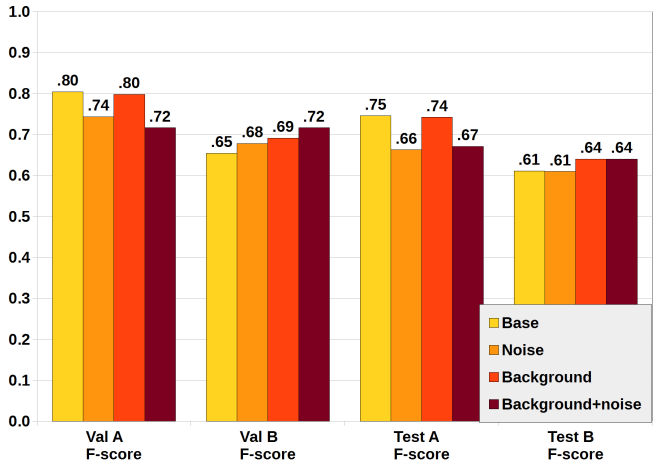


Fig. 8. F-score on *Val* and *Test* sets

performance. The same effect can be seen in figure 9, showing specificity and sensitivity measures of the tests. This enables these neural nest to better perform on non-studio clothing images.

VI. Model architecture experiments

In last series of experiments we attempted to increase classification accuracy by increasing the size of the model at the cost of computational efficiency. For this purpose, we evaluated the accuracy of ensembles of 5 models instead of a single network. Also, we switched the network architecture from SqueezeNet 1.1 to a far larger Resnet-50 architecture [8]. Training of the Resnet-50 network was performed in the same manner as SqueezeNet networks, except that instead of training from scratch, we initiated Resnet-50 networks with weights of a model trained on Imagenet dataset [32] and fine-tuned it. We used 1/100th learning rate for the pretrained layers in comparison to the new classifier layer with random weights. By using the pretrained weights and fine-tuning, we managed to significantly reduce the training time of the networks. Eventually, we tested four model configurations: single SqueezeNet network, ensemble of five independently trained SqueezeNet networks and a similar setup for Resnet networks. Test results for each attribute were presented in figure 10. Again, we also evaluated classifiers accuracy on test set subsets of 200 samples for each attribute with manually corrected labels, which were outlined in figure 11. Sample classification results for Resnet-50 networks we presented in figure 4.

A. Accuracy evaluation

The acquired results show that wrong labels in the datasets significantly disrupted the classification accuracy. Classifiers trained for attributes with highest wrong label ratios achieved lowest accuracy ratios, as expected. Neckline attribute also heavily suffered from large ratio of invalid images. Also, pattern, neckline and color are difficult attributes to classify. Their datasets consist of numerous cases of class overlaps, which is intuitive for color attribute, since there are many

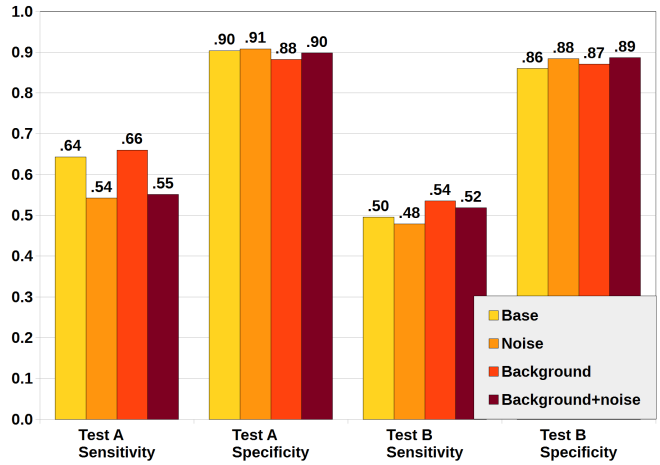


Fig. 9. Sensitivity and specificity on *Test* sets

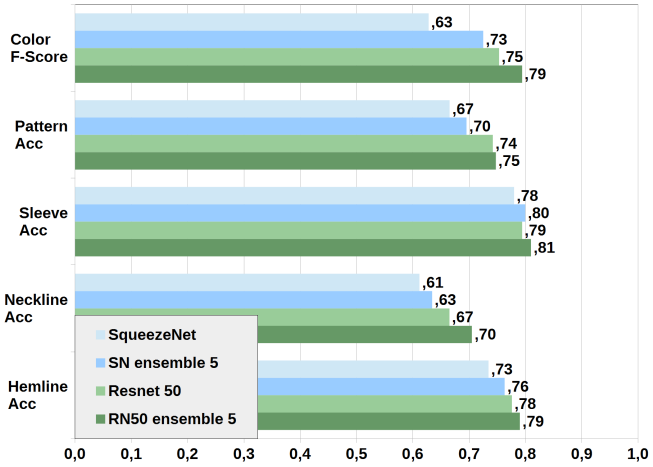


Fig. 10. Classification results acquired on original test set

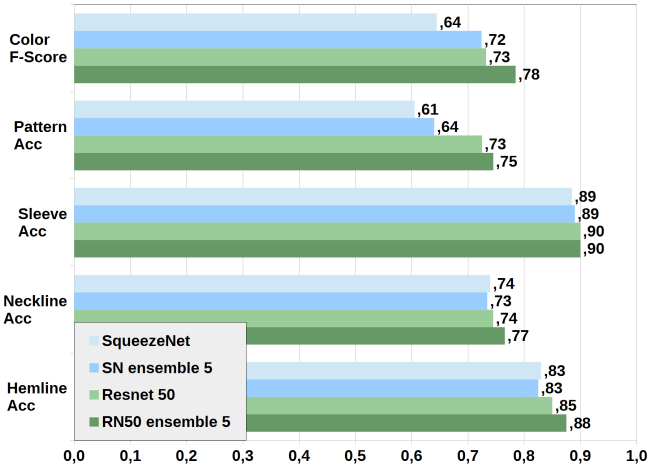


Fig. 11. Classification results acquired on corrected test labels samples

colors that fits more than one class, like turquoise fitting both blue, and green, purple fitting pink and violet or lemon fitting yellow and green. This effect is however common also for other attributes and it adversely influences both training of the models and its precise evaluation. Also, pattern is in fact a multi-class attribute as some clothes have multiple types of patterns. Treating the attribute as single-class therefore limits the accuracy of the models.

Sleeve type and hemline attributes in turn, were simpler to classify. The results were mostly limited by the ratio of wrong labels. We did not observe any spectacular increase in accuracy by using larger networks or ensembles. All models showed a significant increase in accuracy when tested on corrected labels subsets, which confirms high stability of the classifiers. Interestingly, this was the case also for the neckline attribute.

In case of the difficult attributes we observed a significant accuracy gain both by using larger network architecture and employing ensembles. This confirms that larger models perform better in noisy data with high degree of uncertain labels, even in relatively simple task like pattern classification including only several classes. Still, as expected, ensemble of Resnet-50 networks scored highest results for all five attributes and the advantage i most cases justifies the additional computational cost.

B. Performance evaluation

Resnet networks achieve notably higher accuracy results than SqueezeNet, but at the same time suffers from higher computational costs. We benchmarked the two networks as well as the ensembles on a set of 10000 images on a single Titan X Pascal GPU using Caffe framework. We also tested two processing modes: online processing, where short processing time of a single image (latency) is expected, and batch processing, where high throughput is desired. Results where presented in table VI.

VII. Conclusion

We presented a prototype system for automatic detection and classification of clothes that can be potentially applied in automatic product tagging. We achieved clothes detection accuracy for five major types clothes exceeding F-Score of 0.9. The dress classification results ranged from 70% accuracy for neckline to 81% for sleeve type attribute and were mostly limited by the ratio of incorrect labels in the datasets. We presented a background augmentation strategy, which enables increasing classification accuracy for real-life images basing

TABLE VI
Performance evaluation results

Network	Latency	Throughput
SqueezeNet	2.7 ms	422 fps
SqueezeNet ensemble of 5	11.5 ms	140 fps
Resnet50	18.6 ms	98 fps
Resnet50 ensemble of 5	74.1 ms	32 fps

mostly on white-background images, that are most popular in e-commerce. We also verified the significance of the accuracy gain from increasing model architecture to the size of Resnet-50 and using ensembles, even for the quite simple classification task of assigning dress attribute values.

Future work will include correction of the datasets by manual refinement of wrong labels, which can be fairly precisely identified using already trained networks. The datasets will be also extended with more general types of photos, including various shooting angles and body positions. We also plan to develop new configurations of background augmentation with higher ammount of white-backgound images in order to achieve high accuracy for images with varying backgrounds without sacrificing the accuracy for most common images with bright, plain background.

REFERENCES

[1] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-Shop : Cross-Scenario Clothing Retrieval via Parts Alignment and Auxiliary Set," pp. 3330–3337, 2012.

[2] M. Yang and K. Yu, "Real-time clothing recognition in surveillance videos," in *Proceedings - International Conference on Image Processing, ICIP*, 2011, pp. 2937–2940.

[3] H. Chen, A. Gallagher, and B. Girod, "Describing Clothing by Semantic Attributes," *Computer Vision–ECCV 2012*, pp. 609–623, 2012. [Online]. Available: http://chenlab.ece.cornell.edu/people/Andy/publications/ECCV2012{_}ClothingAttributes.pdf

[4] J. Schmidhuber and Gen, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85—-117, 2015.

[5] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] 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 (NIPS)*, 2015.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, 2016, pp. 21–37.

[8] 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.

[9] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," *arXiv:1602.07360*, 2016.

[10] 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 Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, 2015, pp. 1–9.

[11] 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," *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR '15*, pp. 499–502, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=2749318{%}5Cnhttp://dl.acm.org/citation.cfm?doid=2671188.2749318

[12] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, "Where to buy it: Matching street clothing photos in online shops," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 11-18-Dece, pp. 3343–3351, 2016. [Online]. Available: http://acberg.com/papers/wheretobuyit2015iccv.pdf

[13] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan, "Deep domain adaptation for describing people based on fine-grained clothing attributes," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 5315–5324, 2015.

[14] S. Tsogkas, I. Kokkinos, G. Papandreou, and A. Vedaldi, "Deep Learning for Semantic Part Segmentation with High-Level Guidance," no. 2014, pp. 1–11, 2015. [Online]. Available: http://arxiv.org/abs/1505.02438

[15] B. Hasan and D. Hogg, "Segmentation using Deformable Spatial Priors with Application to Clothing," in *Procedings of the British Machine Vision Conference 2010*, 2010, pp. 83.1–83.11. [Online]. Available: http://www.bmva.org/bmvc/2010/conference/paper83/index.html

[16] N. Wang and H. Ai, "Who Blocks Who: Simultaneous clothing segmentation for grouping images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 1535–1542.

[17] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 2691–2699, 2015.

[18] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool, "Apparel classification with style," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7727 LNCS, no. PART 4, 2013, pp. 321–335.

[19] Z. Li, Y. Li, W. Tian, Y. Pang, and Y. Liu, "Cross-Scenario Clothing Retrieval and Fine-grained Style Recognition," *Icpr*, vol. 2, pp. 2919–2924, 2016.

[20] G.-L. Sun, X. Wu, and Q. Peng, "Part-based clothing image annotation by visual neighbor retrieval," *Neurocomputing*, vol. 213, pp. 115–124, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231216307159

[21] Q. Dong, S. Gong, and X. Zhu, "Multi-Task Curriculum Transfer Deep Learning of Clothing Attributes," 2016. [Online]. Available: http://arxiv.org/abs/1610.03670

[22] K.-H. Liu, T.-Y. Chen, and C.-S. Chen, "MVC: A Dataset for View-Invariant Clothing Retrieval and Attribute Prediction," *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pp. 313–316, 2016. [Online]. Available: http://doi.acm.org/10.1145/2911996.2912058

[23] W. Yang, W. Ouyang, H. Li, and X. Wang, "End-to-End Learning of Deformable Mixture of Parts and Deep Convolutional Neural Networks for Human Pose Estimation," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3073–3082, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7780704/

[24] H. Chen, Z. J. Xu, Z. Q. Liu, and S. C. Zhu, "Composite templates for cloth modeling and sketching," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 943–950.

[25] H. N. Ng and R. L. Grimsdale, "Computer graphics techniques for modeling cloth," *IEEE Computer Graphics and Applications*, vol. 16, no. 5, pp. 28–41, 1996.

[26] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan, "Style finder: Fine-grained clothing style detection and retrieval," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 8–13.

[27] J.-C. Chen, C.-F. Liu, and C.-H. Chen, "Deep net architectures for visual-based clothing image recognition on large database," *Soft Computing*, vol. 21, no. 11, pp. 2923–2939, 2017.

[28] S. Hong, B. Roh, K.-H. Kim, Y. Cheon, and M. Park, "PVANet: Lightweight deep neural networks for real-time object detection," *arXiv preprint arXiv:1611.08588*, 2016.

[29] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," *arXiv preprint arXiv:1605.06409*, 2016.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[31] N. McLaughlin, J. M. Del Rincon, and P. Miller, "Data-augmentation for reducing dataset bias in person re-identification," *AVSS 2015 - 12th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2015.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.