Deep Learning for Automated Tagging of Fashion Images

Patricia Gutierrez, Pierre-Antoine Sondag, Petar Butkovic, Mauro Lacy, Jordi Berges, Felipe Bertrand, and Arne Knudson

Amazon.com
{gupatric,pierreas,petarb,lacym,joberges,felipb,knudson}@amazon.com

Abstract. We present 9 deep learning classifiers to predict Fashion attributes in 4 different categories: apparel (dresses and tops), shoes, watches and luggages. Our prediction system hosts several classifiers working at scale to populate a catalogue of millions of products. We provide details of our models as well as the challenges involved in predicting Fashion attributes in a relatively homogeneous problem space.

Keywords: deep learning · image recognition · fashion attributes

1 Introduction

Automatic tagging of products is relevant for online retail applications when dealing with extremely large repositories of products. Given a product image, for instance a dress or a shoe, deep learning models can be generated to predict whether it is a cocktail dress, a black shirt or an stiletto heel. By extracting these tags or attributes from fashion images, queries to the products catalogue can be generated looking for similar or complementary products, produce recommendations for the user, fill missing metadata, and overall provide an improved search experience, all based exclusively on the product image.

In addition, if a product repository is large enough, it becomes impossible to manually audit or populate missing or mislabeled data. Incorrect labeling makes search results not to match what customers are looking for and sales opportunities are lost when products are labeled incorrectly or are undiscoverable. In such cases, the use of machine learning to scale product classification and data quality becomes a strong alternative to manual processes. Using product images as a source of predictions is extremely powerful as images contain a great detail of information about the product, often larger than the description itself.

In this paper, we present deep learning models to detect fashion attributes and populate a catalogue of millions of products. Our system extracts product information from images, automatically and at scale, doing work equivalent to thousands of manual auditors. We have produced 9 classifiers that predict attributes in 4 different categories: apparel, shoes, watches and luggage over 9 different attributes (sleeve type, boot style, heel type, skirt length, sandal style, neck style, watches display type, luggage shell type and number of wheels). As

a result, we have predicted and tagged 18 million images which are available for search and discoverability in our retail platform. In this paper, we present the details of our models, their accuracy, as well as the challenges involved in predicting attributes in the fashion problem space, which is relatively homogeneous with respect to a given product type.

2   CNN for Fashion Attribute Classification

Convolutional Neural Networks (CNNs) have become the leading technique for many image classification tasks [7, 15, 4]. We trained CNN classifiers using both Resnet [4] and GoogleNet [15] architectures. Resnet gives the flexibility to parameterize the number of layers and thus select the size of the network in correspondence to the size of the training set. GoogleNet, on the other hand, facilitates the use of batch normalization and we performed transfer learning using pre-trained models in this architecture.

Table 1 shows the accuracy of our models and the number of tags predicted and used to update our catalogue in Production. Each model predicts an attribute. Details of each model, their labels, and visual examples are shown in the Appendix (Table 5). Labels were extracted from our fashion catalogue and curated by human auditors. We trained most models using a Resnet-50 or Resnet-101 architecture during 100 epochs. Prior to training, we padded the images with white background to make them square with 500x500 pixels. Then, we resized them to 224x224 pixels for training [6]. We used the raw RGB pixel values as input for the convolutional layers and we trained them end-to-end, using random flipping (with 0.5 probability) to augment the training set. We used a learning rate of 0.01 and a stochastic gradient descent optimizer. We tuned the batch size parameter to maximize the number of images per batch without causing memory overflow. We split the training set, using 70%-80% of images for training and the remaining images for validation. We used transfer learning in 3 of our models (*sleeve type*, *neck style* and *display type*). With transfer learning, we improved accuracies by 1%, 2% and 2% respectively. We applied early stopping when the training accuracy started to deviate strongly from the validation accuracy to reduce the overfitting of the models. The base models used to initialize the weights are in-house GoogleNet classifiers (not public) detecting similar fashion tags but not quite tailored to our labels or product domain. In the case of *display type*, we used as base model a GoogleNet network pre-trained on Imagenet [5].

3   Robustness

Even though CNNs have achieved state-of-the-art performance and even human-level performance, they are still subject to anomalies which affect their robustness. Recent studies [8, 10, 11, 14, 16] have shown that a correctly classified image can be changed by introducing small perturbations sometimes even imperceptible to the human eye and yet causing the CNN to misclassify the image into a totally different class. Other studies have shown that it is possible to produce

(a) Resized with Python library. **Prediction: dress**

(b) Resized with Java library. **Prediction: not a dress**
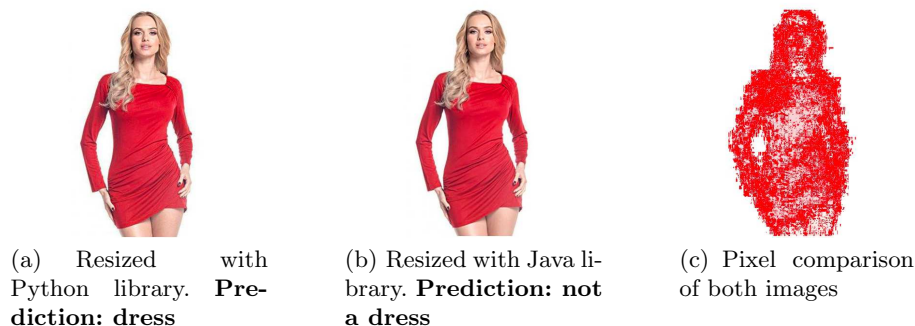
(c) Pixel comparison of both images

Fig. 1. Performance issues related to pixel value perturbations.

Table 2. Accuracy of different models on images resized with different libraries.

Architecture	Accuracy on dataset compressed with Java	Accuracy on dataset compressed with Python
Resnet-18	83%	98%
GoogleNet	93%	99%
Alexnet	98%	98%
GoogleNet with augmentation and base model	98%	98%

3.2 Outliers

We also noticed that images which are incorrectly assigned to the wrong category (e.g. a dress incorrectly stored as a watch) represent outliers for attribute classifiers. If one outlier is presented to the network (e.g. a dress image is presented to a shoe attributes classifier such as *heel type*), the desirable output would be one where all labels have relatively low confidence values. In this way, we could reject unknown cases by simple thresholding. However, the studies mentioned above and our own empirical results show that thresholding over the confidence value is not enough to identify what is unknown. In fact, as shown in Figure 2, for each attribute classifier we can always find an outlier image in our product catalogue which is visually far from the training set and yet it produces a high confidence prediction.

One possible explanation for these anomalies related to outliers is the one described in [12]. Classification models create decision boundaries that partition the data into classification regions. Those classification regions can be much larger than the area occupied by the training set. Outliers far from the decision boundary and deep into the classification region may produce high confidence predictions even when they are far from the training set images. This perspective is further investigated in [2]. The effect of uncalibrated networks producing very high confidence predictions is also studied in [3].

Fig. 2. High confidence misclassifications.

To mitigate this risk, we trained Product Type (PT) classifiers as a pre-filtering step prior to attribute prediction (Table 3). Product type classifiers are binary models that predicts whether an image belongs to a given product type or not. Product types have a hierarchical structure and they represent different categories, such as: shoes, boots, tops, shirts, dresses, watches, bracelets, etc. These models act as a pre-validator for our attributes classifiers.

Table 3. Product Type CNN classifiers.

Product Type	Dataset Size	Number of classes	Accuracy
Luggage	185,036	2	99.48%
Watch	45,880	2	97.30%
Shoe	300,000	2	99.24%
Dress	189,975	2	98.38%
Top	119,567	2	98.56%

Notice that product type classifiers achieve a higher accuracy than attribute classifiers. The problem space of product type classification seems to be easier for a CNN than attribute classification. In other words, it is easier to create a model that is able to distinguish a watch from any other product, than it is to create a model that distinguishes between analogue watches and digital watches. In the latter case, the features that the classifier needs to learn are much more specific and there is more overlap between irrelevant features that are shared across classes. Figure 3 shows a representation of images classified by the watch product type model and images classified by the *display type* attribute model for watches. The images correspond to the training set of each model visualized using the t-SNE [9] algorithm. We can observe that the boundary is better defined and the amount of noise is smaller in the watches PT training set than in the *display type* attribute classifier training set.

Therefore, when performing predictions we first check if the image belongs to the correct product type (e.g. is this image a watch?) and then predict the corresponding attribute (e.g. what type of *display type* does this watch contains?). We have observed that, in addition to product type pre-validation, regularizing the network is effective for ruling out outliers. This is shown in Table 4, which depicts experiments performed on 777 watches images containing outliers. We can
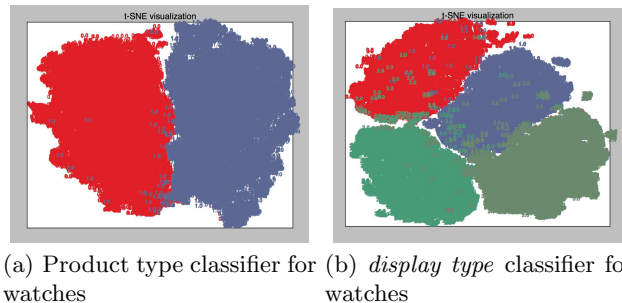
(a) Product type classifier for watches

(b) *display type* classifier for watches

Fig. 3. Visual representation of the training sets.

see how the number of misclassification decreases as the network is regularized (increasing weigh decay and reducing the number of layers). During training, the model was able to fit the training data perfectly and no major accuracy changes were observed by adding regularization. However, when facing an outlier image, the regularized network is able to rule out unknown cases more effectively.

Table 4. Effect of regularizing a network compared to perform PT pre-validation on a noisy set containing outliers. The first column shows the mean confidence value of the predictions. The last three columns show, thresholding by the mean confidence, the percentage of misclassifications, the percentage of misclassifications due to outliers and the overall coverage.

Model	Mean Confidence	Errors	Outlier's errors	Coverage
Resnet-50	0.99	47 (7.3%)	22 (3.4%)	82%
Resnet-18, weight decay 0.3	0.93	37 (5.9%)	19 (3.0%)	79.3%
Resnet-18, weight decay 0.5	0.70	31 (5.5%)	9 (1.6%)	71.9%
Resnet-18, weight decay 1	0.40	23 (5.3%)	2 (0.4%)	55.2%
Resnet-50, with PTD validator	0.99	**21 (3.5%)**	**1 (0.1%)**	75.5%

4   Conclusions

From our research, we found that product images can be a valuable source of information to derive fashion attributes to help the customers find the product they are looking for. We showed that CNN based classifiers can reach accuracies around 90 percent extracting attributes information out of product images. We observed that attribute models have a somewhat reduced training domain which is enclosed to a given category (e.g. shoes, tops, watches, luggages) and provide unreliable predictions when facing an image outside their training scope (outliers). To mitigate such risks, we developed product type classifiers that are able to identify outliers and rule them out. We observed that regularization also provides robustness in these edge cases and in cases where pixel perturbations are introduced in the images as result of pre-processing modifications. As future

work, we want to investigate alternatives to the $softmax$ function to obtain more effective uncertainty estimates. Some recent studies have proposed approaches in that direction, such as [13, 1, 14, 3]. Overall, we have predicted and tagged 18 million attributes in our fashion catalogue with the values extracted from our most confident predictions, increasing their discoverability.

References

1. Bendale, A., Boult, T.E.: Towards open set deep networks. CoRR **abs/1511.06233** (2015)
2. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015), http://arxiv.org/abs/1412.6572
3. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. ICML 2017 (2017)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2016)
5. J. Deng, W. Dong, R.S.L.J.L.K.L., Fei-Fei., L.: Imagenet: A large-scale hierarchical image database. CVPR09 (2009)
6. Johnson, J., Li, F.F., Karpathy, A.: Stanford cs231n: Convolutional neural networks for visual recognition lectures (2016)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
8. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. CoRR **abs/1607.02533** (2016)
9. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008), http://www.jmlr.org/papers/v9/vandermaaten08a.html
10. Mahmood, S., Sruti, B., Lujo, B., K., R.M.: Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. ACM Conference on Computer and Communications Security pp. 1528–1540 (2016)
11. Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. CoRR **abs/1610.08401** (2016)
12. Nguyen, A.M., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: CVPR. pp. 427–436. IEEE Computer Society (2015)
13. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., Hinton, G.E.: Regularizing neural networks by penalizing confident output distributions. CoRR **abs/1701.06548** (2017)
14. Subramanya, A., Srinivas, S., Babu, R.V.: Confidence estimation in deep neural networks via density modelling. CoRR **abs/1707.07013** (2017)
15. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (CVPR) (2015), http://arxiv.org/abs/1409.4842
16. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. CoRR **abs/1312.6199** (2013)

5   Appendix

Table 5. Details of the attributes our classifiers predict.

Attribute	Category	Classes	Examples
sleeve type	Shirts, Blouses, T-Shirts, Dresses	– Long sleeve – Short sleeve – 3/4 sleeve – S leeveless	
boot style	Shoes	– Biker boots – Chelsea boots – Chukka boots – Classic boots – Combat boots – Desert boots – Snow boots – Wellington boots	
display type	Watches	– Analogue – Analogue-Digital – Chronograph – Digital	
heel type	Shoes	– Wedge – Stiletto – Kitten Heel – Cone Heel – Western Heel – Block Heel – Louis Heel – Flat	
shell type	Luggage	– Hard – Soft	

Attribute	Category	Classes	Examples
skirt length	Dresses, Skirts	− Mini − Knee-Long − Midi − Maxi	
sandal style	Shoes	− Ankle-Strap − Espadrille − Flatform − Gladiator − Platform − Slingback − T-Bar	
neck style	Dresses	− Asymmetric − Collared − Boat Neck − Round Neck − Turtleneck − V-Neck − Halterneck − One-Shoulder − Sweetheart − Square − High Neck − Off the Shoulder	
number of wheels	Luggage	− 0 − 2 − 4	