Leveraging Weakly Annotated Data for Fashion Image Retrieval and Label Prediction

Charles Corbière[1], Hedi Ben-Younes[1,2], Alexandre Ramé[1], and Charles Ollion[1]

[1]Heuritech, Paris, France
[2]UPMC-LIP6, Paris, France
{corbiere,hbenyounes,rame,ollion}@heuritech.com

Abstract

In this paper, we present a method to learn a visual representation adapted for e-commerce products. Based on weakly supervised learning, our model learns from noisy datasets crawled on e-commerce website catalogs and does not require any manual labeling. We show that our representation can be used for downward classification tasks over clothing categories with different levels of granularity. We also demonstrate that the learnt representation is suitable for image retrieval. We achieve nearly state-of-art results on the DeepFashion In-Shop Clothes Retrieval and Categories Attributes Prediction [12] tasks, without *using the provided training set.*

1. Introduction

While online shopping has been an exponentially growing market for the last two decades, finding exactly what you want from online shops is still not a solved problem. Traditional fashion search engines allow consumers to look for products based on well chosen keywords. Such engines match those textual queries with meta-data of products, such as a title, a description or a set of tags. In online luxury fashion for instance, they still play an important role to address this customer pain point: 46% of customers use a search engine to find a specific product; 31% use it to find the brand they're looking for [1]. However, those meta-data informations may be incomplete, or use a biased vocabulary. For instance, a description may denote as "marinière" a long sleeves shirt with blue/white stripes. It then appears crucial for online retailers to have a rich labeled catalog to ensure good search. Moreover, these search engines don't incorporate the visual information of the image associated to the product.

[1]http://www.mckinsey.com/business-functions/marketing-and-sales/our-insights/the-opportunity-in-online-luxury-fashion

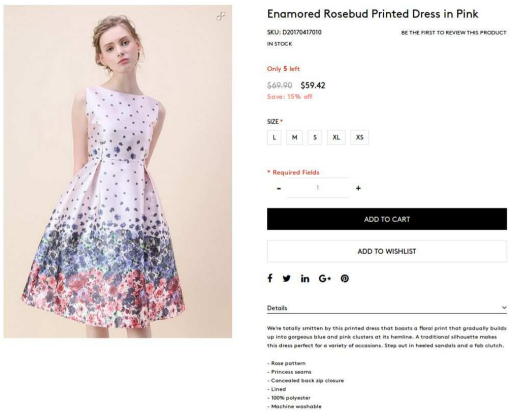

Figure 1. Our dataset is composed of images and a few associated text descriptors such as their title and their description

Computer vision for fashion e-commerce images has drawn an increasing interest in the last decade. It has been used for similarity search [20, 11, 21, 18], automatic image tagging [10, 2], fine-grained classification [5, 12] or N-shot learning [1]. In all of these tasks, a model's performance is highly dependent on a visual feature extractor. Using a Convolutional Neural Network (CNN) trained on ImageNet [4] provides a good baseline. However, there are two main problems with this representation. First, it has been trained on an image distribution that is very far from e-commerce, as it has never (or rarely) seen such pictures. Second, the set of classes it has been trained on is different from a set of classes that could be meaningful in e-commerce. A useful representation should separate different types of clothing (*e.g.* a skirt and a dress), but it should also discriminate between different lengths of sleeves for shirts, trouser cuts, types of handbags, textures, colors, shapes,...

Our goal is to learn a visual feature extractor designed for e-commerce images. This representation should:

- encode multiple levels of visual semantics: from low level signals (color, shapes, textures, fabric,...) to high

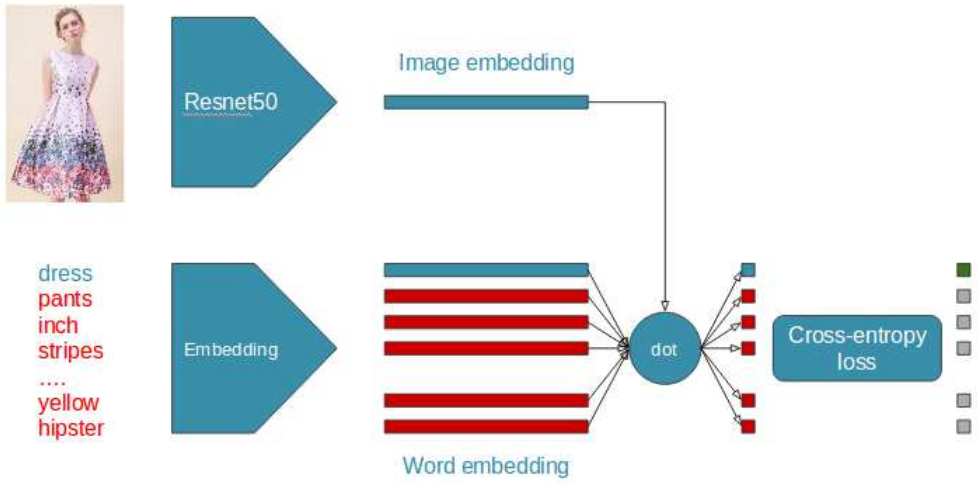

Figure 2. Training of our model: predict one label, picked from the bag-of-words description, from an image. Both image and words are embedded before being coupled in a dot product. We output a probability for each word in the vocabulary.

level information (style, brand),
- be separable over visual concepts, so we can train very simple classifiers over clothing types, colors, attributes, textures, *etc.*,
- provide a meaningful similarity between images, so we can use it in the context of image retrieval.

To these ends, we train a visual feature extractor on a large set of weakly annotated images crawled from the Internet. These annotations correspond to the textual description associated to the image. The model is learned on a dataset at *zero* labeling cost, and is exclusively constituted of data points extracted from e-commerce websites. Our main contribution is an in-depth analysis of the model presented in [9], through applications to fashion image recognition tasks such as image retrieval and attribute tagging. We also improved the method by upgrading the CNN architecture and dealt with multiple languages, mainly English and French. In Section 2, we explain the model, how we handle noise in the dataset, as well as some implementation details. In Section 3, we provide results given by our representation on image retrieval and classification, over multiple datasets. Finally in Section 4, we conclude and go over some possible improvement tracks.

2. Learning Image and Text Embeddings with Weak Supervision

One major issue in applied machine learning for fashion is the lack of large clothing e-commerce datasets with a rich, unique and clean labeling. Some very interesting work has been done on collecting datasets for fashion [11, 12]. However, we believe it is very hard to be exhaustive in describing every visual attribute (pieces of clothing, texture, color, shape, *etc.*) in an image. Moreover, even if this labeling work could be perfectly carried, it would come at very high cost, and should be manually done each time we wanted to add a new attribute. A possible source of annotated data is the e-commerce website catalogs. They provide a great amount of product images associated with descriptions, such as the one in Figure 1. While this description contains information about the visual concepts in the image, it also comes with a lot of noise that could harm the learning.

We explain now the approach we used to train a visual feature extractor on noisy weakly annotated data.

2.1. Weakly Supervised Approach

Learning with noisy labeled training data is not new to the machine learning and computer vision community [6, 17, 19]. Label noise in image classification [22] usually refers to errors in labeling, or to cases where the image does not belong to any of the classes, but mistakenly has one of the corresponding labels. In our setting, in addition to these types of noise, there are some labels in the classes vocabulary that are not relevant to any input. Text descriptions are noisy as they contain common words (*e.g.* 'we', 'present'), subjective words (*e.g.* 'wonderful') or non visual words (*e.g.* 'xl', 'cm'), which are not related to the input image. As we don't have any prior information on which labels are relevant and which are not, we keep the preprocessing of textual data as light as possible.

2.2. Model

Our work builds upon the one presented in [9], which we explain in this section. The model's training scheme is exposed in Figure 2

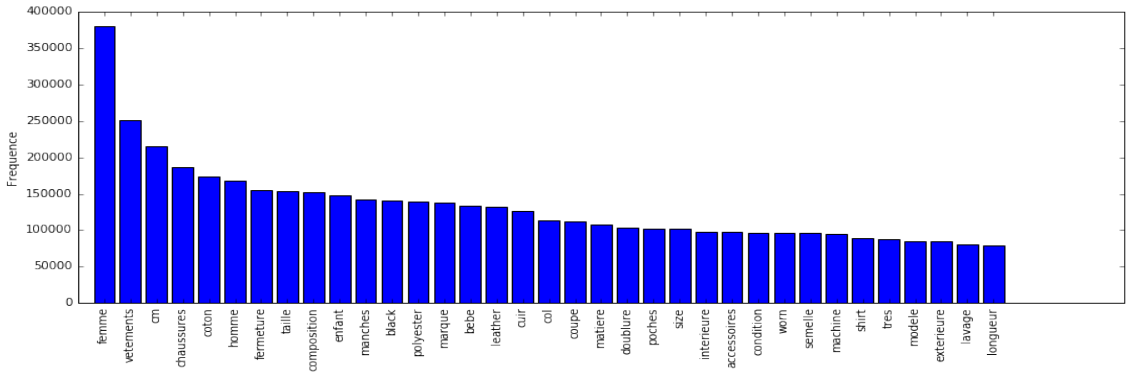

Figure 3. Most frequent labels in training dataset.

Let $x \in \mathcal{I}$ be an image, and $y \in \{0,1\}^K$ the associated multi-label vector, such that $\forall k \in [1,K], y^k = 1$ if the k-th label of the vocabulary is true for the image x. We use a CNN to compute a visual feature $z = f(x,\theta) \in \mathbb{R}^I$, where θ are the weights of our convolutional neural network. This image embedding is given to a classification layer:

$$\hat{y} = softmax\left(W^T z\right) \tag{1}$$

where $W \in \mathbb{R}^{I \times K}$. Note that for all $k \in [1,K]$, the column vector in $w_k = W[:,k]$ corresponds to the embedding of the k-th word in the vocabulary.

2.3. Label imbalance management

As seen on Figure 3, the distribution of words in our dataset is highly unbalanced. Due to our minimal preprocessing, we observe a high frequency for some non-visual words such as "xl", "cm" or "size" as they appear very frequently in descriptions. Many examples contain those non-visual words that our model would be asked to predict, which is likely to harm the training.

To overcome this issue, and as it was done in [9], we perform *uniform* sampling. Specifically, during training, we sample uniformly a word w from the vocabulary. We then randomly choose an image x whose bag-of-words contains w and we try to predict w given x.

2.4. Loss

As we want to predict one label among a vocabulary K for each image, we use the cross-entropy loss. It minimizes the negative sum of the log-probabilities over all positive labels

$$L(\theta, W, \mathcal{D}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_n^k \log \frac{\exp\left(w_k^T f(x_n,\theta)\right)}{\sum_{i=1}^{K}\exp\left(w_i^T f(x_n,\theta)\right)} \tag{2}$$

2.5. Implementation details

Negative sampling
We operate in a context where the vocabulary can be of arbitrary size. Computing probabilities for all those classes for each sample can be very slow. Negative sampling [16] is one way of addressing this problem. After selecting a positive label for an image sample, we randomly draw N_{neg} negative words within the vocabulary. We compute the scores and the softmax only over those chosen words.

Learning
We trained our model with stochastic gradient descent (SGD) on batch of size 20. We consider that an epoch is achieved when the model saw a number of images equivalent to $1/10$ of the dataset size, which is approximately 1.3M images in total. After each epoch, we compute a validation error based on a held-out validation set. The initial rate was set to 0.1 and divided by 10 after 10 epochs without improvement. We stop the training after 20 epochs without improvements on our validation dataset. We use the ResNet50 architecture [7] for the visual feature extractor $f(x,\theta)$, with pre-trained weights on ImageNet. Because the last layer has been initialized randomly, we start by learning only the last layer W for 20 epochs, and then we fine-tune the parameters θ in the CNN.

2.6. Training dataset

We built a dataset of about 1,3M images and their associated labels from multiple e-commerce website sources, mostly French, English and Italian. We crawled most of the time one image per product, except when multiple images where available. In that case, we consider them as four different samples with same associated bag of labels.

For each source, we select the relevant fields to keep (title, category name, description,...) and concatenate them. After lower-casing and removing punctuation, we use the RegexpTokenizer provided by NLTK [13] to get a list of words. We remove stopwords, frequent non-relevant words

Figure 4. Comparison between visual similar images from DeepFashion In-Shop dataset according to our weakly learned visual features and ImageNet based visual features. Our representation seems more robust to human pose (on the left) and successfully captured fine-grained concepts such as stripes (on the right).

(name of the website, 'collection', 'buy',...) and non alphabetic words. Our final dataset is a list of product images, associated to their respective bag-of-words obtained by the previous preprocessing.

We have deliberately kept preprocessing as minimal as possible, so it is easy to scale to many sources. Thus, we need our model to adapt to this noise in the data. After preprocessing and aggregating the multiple sources, our final vocabulary is composed of 218,536 words. We chose to restrain the vocabulary to the 30,000 most frequent words. The average number of labels per sample is 26,88. We split our dataset into a training and a validation dataset. The validation set is made of the same labels as the training dataset and represent 0.5% of the total size.

3. Experiments and evaluation

After learning the representation on our large weakly annotated dataset, we want to evaluate this representation. To what extent is this representation useful for tasks such as garment classification, attribute tagging or image retrieval ?

3.1. Evaluation datasets

We evaluate our representation on 5 datasets: two public datasets (DeepFashion) used for tagging and image retrieval; three in-house datasets used respectively for category classification, fine-grained classification and image retrieval.

DeepFashion Categories and Attributes Prediction evaluates the performance on clothing category classification, and on attribute prediction (multi-labelling). It contains 289,222 images for 50 clothing categories and 1,000 clothing attributes. While an image can only be affected to one class, it can be associated to multiple labels. The average number of labels for an image is 3.38. For each image in train and test sets, we select a crop available from a ground truth bounding box.

DeepFashion In-Shop Retrieval contains 7,982 clothing items with 52,712 images. 3,997 classes are for training (25,882 images) and 3,985 items are for testing (28,760 images). The test set is composed of a query set and a gallery set, where query set contains 14,218 images of 3,985 items and database set contains 12,612 images of 3,985 items. As in the Categories and Attributes Prediction benchmark, we cropped each image using a ground truth bounding box.

ClothingType We have labeled a dataset with 18 classes, each one corresponding to a garment type (*e.g.* bag, dress, pants, shoes, ...). This in-house dataset contains approximately 736,000 images.

HandBag In addition to the previous dataset, this in-house dataset focuses on bags for fine-grained recognition. Here, the differences between classes are more subtle: bucket bag, doctor bag, duffel bag, etc... It contains 3,060 samples within 13 classes, each one corresponding to a specific type of handbag.

Dress Retrieval This in-house similarity dataset was gathered by crawling an e-commerce website. We collected a list of sets of images, each corresponding to the same item. We used an image classifier to filter out all non-dress items. The final dataset contains 9,009 items for training (20,200 images) and 1,001 items for testing. On this dataset, we keep only images where clothing are worn on humans.

3.2. Image retrieval

In this task, given a query image containing an item, we aim at retrieving images that contain the same item. To do so, we compute the score between two images using the cosine similarity between their representation. For a given query image, we sort all gallery images in decreasing order of similarity, and evaluate our retrieval performance using top-k retrieval accuracy, as in [12, 23]. For a given test query image, we give the model a score of 1 if an image of the same item is within the k highest scoring gallery images, 0 else. We adopt this metric for both our image retrieval datasets (DeepFashion In-Shop Retrieval and Dress Retrieval).

	Category		Texture		Fabric		Shape		Part		Style		All	
	top-3	top-5	top-3	top-5	top-3	top-5	top-3	top-5	top-3	top-5	top-3	top-5	top-3	top-5
WTBI [3]	43.73	66.26	24.21	32.65	25.38	36.06	23.39	31.26	26.31	33.24	49.85	58.68	27.46	35.37
DARN [8]	59.48	79.58	36.15	48.15	36.64	48.52	35.89	46.93	39.17	50.14	66.11	71.36	42.35	51.95
FashionNet [12]	82.58	90.17	37.46	49.52	**39.30**	**49.84**	39.47	48.59	**44.13**	**54.02**	**66.43**	**73.16**	**45.52**	**54.61**
Lu et al.* [14]	**86.72**	92.51	-	-	-	-	-	-	-	-	-	-	-	-
Weakly	86.30	**92.80**	**53.60**	**63.20**	39.10	48.80	**50.10**	**59.50**	38.80	48.90	30.50	38.30	23.10	30.40

*Attribute scores not tested.

Table 1. Performance of category classification and attribute prediction on DeepFashion dataset

In Figure 5, we show the results on top-k retrieval accuracy on DeepFashion In-shop Retrieval dataset, for multiple values of k. FashionNet corresponds to the model presented in [12], and HDC+Contrastive is the model in [23]. We denote by [F] (resp. [C]) models that use the full image (resp. an image cropped on the product) to compute retrieval scores. We provide the ImageNet baseline both [F] and [C] models, where we use as feature extractor the penultimate layer of a CNN trained on ImageNet. We would like to emphasize on the fact that our Weakly model, as well as the ImageNet baseline, *do not use the training set* of DeepFashion In-shop Retrieval, unlike HDC+Contrastive and FashionNet.

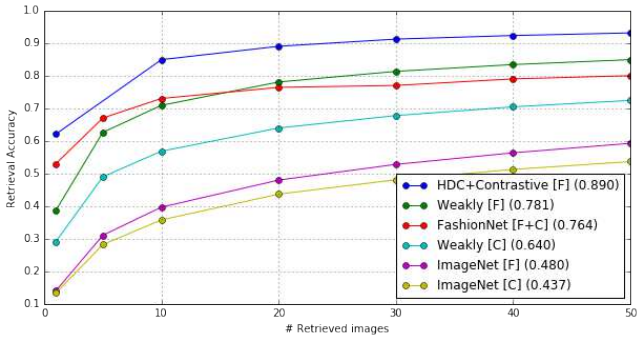

Figure 5. Retrieval accuracy for top-k (k=1,5,10,20,30,40,50). We give the top-20 retrieval accuracy between brackets for each model in the caption.

First, we note that not using bounding boxes for our ImageNet baseline or our Weakly model considerably increase accuracy. Our intuition is that human models in the DeepFashion In-shop dataset often wear the same ensemble of items together, meaning for one shirt item considered for instance, the human model would be wearing the same pants and shoes on all item's image. As a consequence of this bias, it seems easier to evaluate similarity on a ensemble of clothings than on a single clothing on this dataset.

Our Weakly model without crop performs as well as FashionNet, and even outperforms it when $k \geq 20$: considering top-20 retrieval accuracy, it predicts the correct item 78,1% of the time, against 76,4% for FashionNet. Besides, in both the [F] and [C] setups, our Weakly model improves over the ImageNet baseline (from 48% to 78.1% for [F], and from 43.7% to 64.0% for [C]). This validates our hypothesis that our model has learned a specific e-commerce representation. In Figure 4, we show an example of a query image, its top-5 similar images according to our weakly learned visual features, and its top-5 similar images according to ImageNet based visual features. As we can see, the similarity encoded by network trained on ImageNet brings together products that are on a same coarse semantic concept, while our representation encodes a more precise and rich closeness, which is based not only the image type, but also on their shape, texture, and fabric. Plus, our representation seems less dependent to human model's pose.

On our in-house dress retrieval dataset, we also observed that the Weakly model improved over ImageNet Model on retrieval accuracy. The Weakly model obtained a top-20 retrieval accuracy of 83,71%, against 65,65% for the ImageNet model. Once again, we point out that we do not perform any training on the retrieval task of this dataset.

3.3. Tagging

We conducted multi-class classification and multi-labelling experiments to assess the quality of our visual representation on transfer learning. On the public DeepFashion Categories dataset, we pre-computed images representation using our Weakly image feature extractor on image crops. Then, we train a simple classifier using a fully-connected layer followed by a softmax activation function. The results are shown on the Table 3.1, at the column Category. With this simple classifier, our results are on par with the state-of-art model by Lu et al. [14].

On DeepFashion Attributes, we train a fully-connected layer with a sigmoid output and a binary cross-entropy loss. As we can see in Table 3.1, our model significantly improves over previous state-of-the-art on textures and shape labels top-k recall. However, part and style attributes seem more difficult to separate for our Weakly representation. This might be due to the fact that texture-like and shape-like labels are more represented than part and style words in the large weak dataset. This would require further investigation.

We carried out experiments on our in-house Clothing-

Figure 6. t-SNE map of 1,000 image samples from DeepFashion Categories dataset based on our Weakly image features extractor. We can identify some local subcategories, such as colorful dresses (a), black pants (b), stripes (c), checked (d) or printed shirt (e).

	bag	belt	body	bra	coat	combi	dress	eyewear	gloves	hat	neckwear	pants	shoes	shorts	skirt	socks	top	underpants
ImageNet	99.63	98.02	98.20	99.27	99.00	96.01	98.68	99.93	**99.99**	99.45	99.18	99.46	99.87	99.23	98.67	99.35	98.67	99.5
Weakly	**99.86**	**99.46**	**99.08**	**99.67**	**99.31**	**98.11**	**99.13**	**99.99**	99.97	**99.73**	**99.33**	**99.56**	**99.93**	**99.32**	**99.21**	**99.83**	**99.26**	**99.67**

Table 2. AUC classification score for clothing categories

	backpack	baguette	bowling bag	bucket bag	doctor bag	duffel bag	hobo bag	luggage	clutch	saddle bag	satchel	tote	trapeze
ImageNet	95.15	87.63	90.42	94.35	90.99	87.97	92.73	**87.65**	96.52	91.77	88.58	96.77	92.11
Weakly	**95.94**	**91.85**	**92.13**	**94.87**	**91.59**	**90.12**	**95.19**	86.96	**97.24**	**93.12**	**91.45**	**97.64**	**93.61**

Table 3. AUC classification score for fine-grained type of bags

Type dataset where images are annotated according to their clothing category (such as bags, shirt, dress, shoes, *etc.*). Table 3.3 shows the improvement on AUC scores over the ImageNet model for each of the clothing categories using our new representation. This indicates that our training scheme was able to learn discriminative features for garment classification.

Finally, we now focus on a fine-grained recognition task. The HandBag dataset contains images annotated with their specific type of bag. In this dataset, the differences between classes are more subtle than in the ClothingType dataset. The training and evaluation are the same as for the previous experiment. As in the previous experiment, we improved AUC scores for nearly each type of bags (see Table 3.3).

3.4. Exploratory visualization using t-SNE

To obtain some insight about our Weakly representation, we applied t-SNE [15] on features extracted using our Weakly feature extractor. We did this for 1,000 images from DeepFashion Categories test set. Figure 6 shows full map and some interesting close-ups. On top left (a), we can see a cluster of dresses sub-divised into multiple sub-clusters corresponding to different colors. The cluster (b) shows a focus on black pants. In the zone (c), we can easily see that the model gathered images containing stripes, and it seems like it has separated tops from dresses inside this cluster (with large striped sweaters on top). Checked clothings are grouped in cluster (d), while printed t-shirts are represented in cluster (e). This plot shows that our representation is able to group together concepts that are close in terms of clothing type, texture, color and style.

4. Discussion and Future Work

We presented in the future a method to learn a visual representation adapted to fashion. This method has the major advantage to overcome the issue of finding a large and clean e-commerce dataset. The results shows clear improvements compared to a visual representation trained on ImageNet, improving performance on multiple tasks such as image retrieval, classification and fine-grained recognition.

In the future, we would like to investigate on the possibility to better train our visual feature extractor using an external knowledge base of textual concepts.

Acknowledgments

The authors would like to thank all the Heuritech team for providing an efficient network infrastructure. We'd especially like to thank Pierre Dubreuil for its help on transfer learning evaluation tasks.

References

[1] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *NIPS*, pages 181–189. Curran Associates, Inc., 2010.

[2] 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.

[3] H. Chen, A. C. Gallagher, and B. Girod. Describing clothing by semantic attributes. In A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *ECCV (3)*, volume 7574 of *Lecture Notes in Computer Science*, pages 609–623. Springer, 2012.

[4] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

[5] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style recognition and retrieval. In *IEEE International Workshop on Mobile Vision*, Portland, OR, 2013.

[6] B. Frnay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[8] J. Huang, R. S. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. *CoRR*, abs/1505.07922, 2015.

[9] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache. Learning visual features from large weakly supervised data. *CoRR*, abs/1511.02251, 2015.

[10] Y. Kalantidis, L. Kennedy, and L.-J. Li. Getting the look: Clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, ICMR '13, pages 105–112, New York, NY, USA, 2013. ACM.

[11] 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. In *ICCV*, pages 3343–3351. IEEE Computer Society, 2015.

[12] Z. Liu, P. Luo, S. Qiu, X. Wang, X. Tang, undefined, undefined, undefined, and undefined. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 00:1096–1104, 2016.

[13] E. Loper and S. Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[14] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. S. Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. *CoRR*, abs/1611.05377, 2016.

[15] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[17] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1196–1204. Curran Associates, Inc., 2013.

[18] D. Shankar, S. Narumanchi, H. A. Ananya, P. Kompalli, and K. Chaudhury. Deep learning based large scale visual recommendation and search for e-commerce. *CoRR*, abs/1703.02344, 2017.

[19] S. Sukhbaatar and R. Fergus. Learning from noisy labels with deep neural networks. *CoRR*, abs/1406.2080, 2014.

[20] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393. IEEE Computer Society, 2014.

[21] X. Wang, Z. Sun, W. Zhang, Y. Zhou, and Y.-G. Jiang. Matching user photos to online products with robust deep features. In J. R. Kender, J. R. Smith, J. Luo, S. Boll, and W. H. Hsu, editors, *ICMR*, pages 7–14. ACM, 2016.

[22] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, pages 2691–2699. IEEE Computer Society, 2015.

[23] Y. Yuan, K. Yang, and C. Zhang. Hard-aware deeply cascaded embedding. *CoRR*, abs/1611.05720, 2016.