Fashion Attributes Classification by Convolutional Neural Network

Hongli Deng
Softlines
honglden@amazon.com

Rui Luo
Softlines
luorui@amazon.com

Quoc-Anh Vu
Softlines
quocanhv@amazon.com

Shuai Lu
Softlines
lushuai@amazon.com

Lara Lu
Softlines
lutianyi@amazon.com

Gabriel Blanco
Softlines
saldanag@amazon.com

Karolina Tekiela
Softlines
ktekiela@amazon.com

Abstract

We used a deep convolutional neural network to automatically attribute Softlines ASINs. Classifying fashion is particularly challenging, as style attributes are often subjective and images vary depending on the vendor. The network GoogLeNet[2] plus batch normalization[1], also called Inception_BN in MXNet[6] was adopted to train the image classifiers in our domain. To pursue the goal of high classification accuracy and fast, iterative training, a base model called "Softlines Base CNN" was trained using 18 million images and text information associated with the images. Fast and effective final model training was achieved by refining the base model using relatively small numbers of images for each attribute class. Finally, five dress models were trained, evaluated and deployed in production. In this paper, we not only discuss the full-fledged image classification system, but also demonstrate a thorough understanding of training deep neural networks.

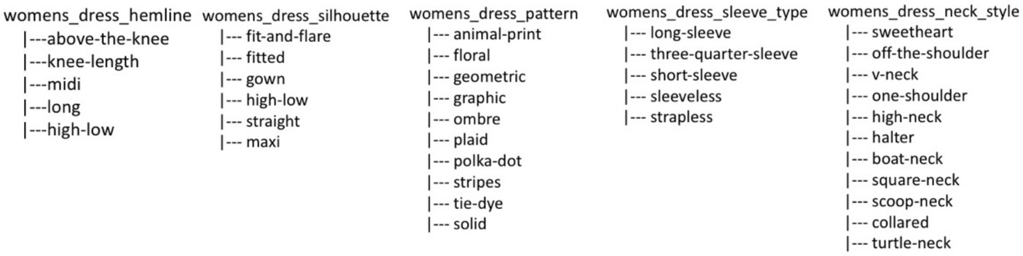
1 Challenges

1.1 Softlines search and discovery

Search is the primary discovery mechanism for customers looking for Softlines products on Amazon, contributing 48.42% of worldwide attributed sales in 2016. Amazon customers tend to use keyword search to find what they want: 95% of all Softlines in-category search traffic on Amazon can be attributed to keyword searches. However, our key search metrics and user studies show customers are having trouble finding what they are looking for. Customers are overwhelmed by Amazon's selection and frustrated when they see incorrect results. In order for the A9 search engine to display the correct ASINs that match each search, ASINs must be tagged and indexed accurately. Currently, the keywords for Softlines ASINs are applied manually and inconsistently by vendors and internal teams. Not only is this method difficult to scale, but it lends itself to error. Vendors often try to boost their products in search by "spamming" their products with incorrect keywords, and there are no guidelines to ensure consistent product labels. This results in a poor customer search experience with irrelevant results.

1.2 The challenge of dealing with machine learning on image data

In order to solve this problem, our team built machine learning models to generate ASIN attributes. The intention is for these keywords to be used for search indexing and matching. For high-level categorical classification, such as whether a product is a dress or a pant, we used text-based models. For specific attributes, whose information are hard to be extracted from texts, such as dress silhouette, pattern, hemline, sleeve type, and neck style, we applied image-based classification, or “Visual Attributes Classification”. Currently, we are focusing on the dress category, as it is a strategic growth category for the Softlines business with a complex variety of attributes. The five most relevant women dress classifiers are defined in (Figure 1).



```
womens_dress_hemline
  |---above-the-knee
  |---knee-length
  |---midi
  |---long
  |---high-low

womens_dress_silhouette
  |--- fit-and-flare
  |--- fitted
  |--- gown
  |--- high-low
  |--- straight
  |--- maxi

womens_dress_pattern
  |--- animal-print
  |--- floral
  |--- geometric
  |--- graphic
  |--- ombre
  |--- plaid
  |--- polka-dot
  |--- stripes
  |--- tie-dye
  |--- solid

womens_dress_sleeve_type
  |--- long-sleeve
  |--- three-quarter-sleeve
  |--- short-sleeve
  |--- sleeveless
  |--- strapless

womens_dress_neck_style
  |--- sweetheart
  |--- off-the-shoulder
  |--- v-neck
  |--- one-shoulder
  |--- high-neck
  |--- halter
  |--- boat-neck
  |--- square-neck
  |--- scoop-neck
  |--- collared
  |--- turtle-neck
```

Figure 1. Women dress classifiers and their attributes.

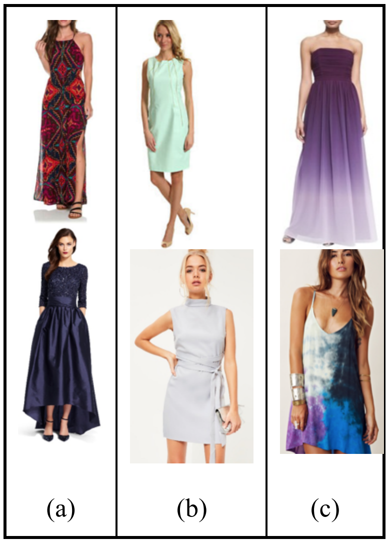


Figure 2. Intrinsically hard to classify, even for a human a) Maxi vs. Gown, b) Straight vs. Fitted, c) Ombre pattern vs. Tie-Dye pattern (both feature gradual color gradation)

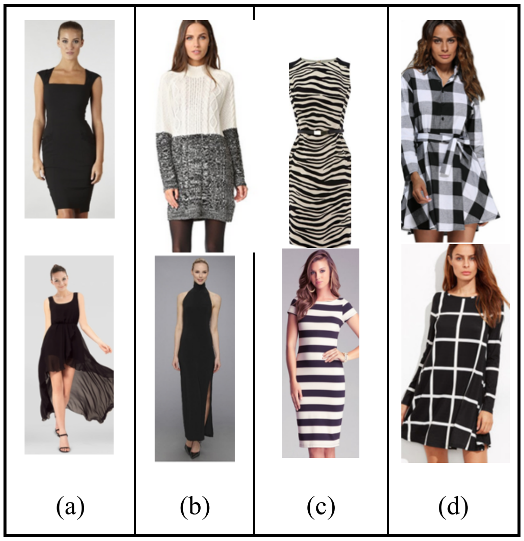


Figure 3. Subtle “between class” differences. a) Square neck vs. Scoop neck, b) High neck vs. turtle neck, c) Animal pattern vs. Stripe pattern, d) Plaid pattern vs. Geometric pattern.

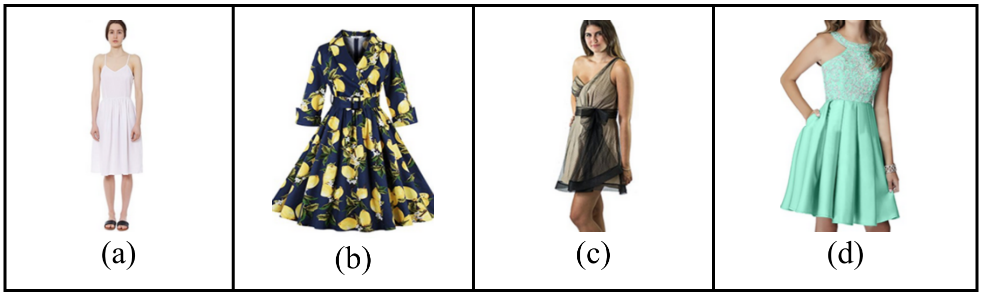


Figure 4. This demonstrates the large class variance within the same class type: Fit & Flare dress. a) with human model full body, b) without human model, c) with human model, half leg, some angles, d) with model, half leg, no head, with arm gestures.

This was a challenge to solve with machine learning: 1) We needed human tags for our machine learning models, but the ground truth labels were hard to get. We learned that fashion domain knowledge and business insights are necessary when defining and classifying attributes (Figure 2). 2) Subtle "between class" differences exist from the image perspective (Figure 3). 3) There are huge variances between vendor images, such as the model gesture and view point change. (Figure 4). When designing the classifier, we took all these factors into account. We balanced the importance of each attribute to customers and the business, as well as technical practicality.

2 Visual Attributes Classification

2.1 Convolutional neural network for image classification

The use of a computer program to automatically classify products has been one of the main use-cases for computer vision technology. Starting from 2012, deep learning technology has been demonstrated to be very powerful in image recognition. Since then Convolutional Neural Networks (CNN) have become increasingly popular for image-based classification. Using a Convolutional Neural Network (CNN) to classify images greatly increases the classification accuracy and outperforms traditional methods. This breakthrough was mainly due to the availability of stronger computational power brought by GPU (Graphics Processing Unit), and the powerful representation capability of the deep neural networks. More and more new network structures were designed over time, from early stage LeNet[4] to the more recent VGG Net[3], GoogleNet[1] and ResNet[5].

2.2 Deep neural network design

We adopted GoogleNet with batch normalization [1] as our neural network for image classification. We call this network Inception-BN. The main reasons for choosing this network were: 1) The Inception module [2] uses a rather deep network which increases the representation power. The computational complexities are controlled by using the Inception model as network building blocks. The model also limits the number of network connections by restricting most of the connection within the modules. This reduces the total number of parameters and achieves significant quality gains with modest increases of computational requirements. 2) Adding batch normalization to the network improves the speed of training by making the forward and back propagation more effective. We directly used the Inception-BN network provided by the Amazon Deep MXNet.

2.3 Train CNN with Base model + Refined model

The cornerstone of our approach was to train a Convolutional Neural Network (CNN) from scratch to classify Softlines product image data. To build such a network, we adopted the base model plus refining model methodology. An overall base model for all Softlines images was first trained. Upon this base model, various refined, customized models were further built. We adopt this methodology for a variety of reasons. First, it typically requires millions of images to train a network as complicated as Inception-BN from scratch, but our available human-tagged ground truth samples are relatively limited. This leads to an under-fitting problem. As a result, the initially randomized parameters cannot be tuned close enough to a global optimal location. Second, even if we have enough human tagged samples, the time for training a new model would be long, this will decrease our parameter tuning speed. Finally, most of the visual attributes models currently in production are from a CNN that was created by adapting a "pre-trained" CNN, that is, one that has been previously trained for another type of image recognition task.

To justify the reason of using refined model, we first directly use the model pre-trained by others using the ImageNet dataset. This is a 1000-class image recognition problem in which the classes correspond a variety of animals, plants and other phenomena found in the natural and man-made worlds. Models that have been trained for this task are available online for free. However, we quickly identified problems with using this model. This model was trained for the ImageNet task which involves discriminating between a diverse range of objects including cars, boats, horses, flowers, buildings and clouds. The features extracted by this CNN have been optimized for those objects. Our experimental results show it didn't help us in terms of training accuracy (see section 3 for explanation).

We realized that a new feature extractor that is more suited to fashion image discrimination tasks is needed. This was the primary motivation for creating our own base model, called the Softlines Base CNN. We believed that it could be used as a starting point to train more accurate visual attributes classifiers using less training data. The initial challenge of training our own Softlines Base CNN was to find a sufficiently large training data set. Fortunately, with Amazon scale, we were able to find enough images using text labels derived from product titles.

2.4 Train the Base model: Softlines Base CNN

We trained the Softlines Base CNN using 18 million images (each is associated with one ASIN) with 8687 total labels. The labels were generated by checking the word frequencies in the title of each of the 18 million ASINs. Three base models were trained individually as shown in table 1, in which Softlines_Top500 means there are 500 classes and so forth. The main purpose of this training is not to do the final classification, instead, it moves the network parameters to a good location from which further fine tuning could be conducted on. During the fine tuning, the last full connected layer of the network is replaced with the number of nodes that is the same as the number of classes in the fine-tuning classifier.

Table 1. Softlines CNN base model experimental datasets

Dataset	**# training images**	**# validation images**	**# labels**
Softlines_2M	Randomly sampled 2 million images	300k	All 8,687 words from tittle
Softlines_Top500	All 14.4 million images	All 3.6 million	Most frequent 500 words from title
Softlines_Top2000	All 14.4 million images	All 3.6 million	Most frequent 2000 words from title

2.5 Distributed Training on EC2 cluster

We used Amazon AWS CloudFormation to build and deploy the distributed training environment. The CloudFormation Deep Learning uses the Amazon Deep Learning AMI to launch a cluster of Amazon EC2 instances and other AWS resources needed to perform distributed deep learning. From the CloudFormation console, we defined one g2.8xlarge instance as the master node and six extra g2.8xlarge instances as workers. The main reason for using cluster-based training was to improve the training speed and make development iteration and parameter tuning faster.

3 Experiments and Performance Evaluation

3.1 Refine Model Datasets

After acquiring our base models, we focused on five types of attributes for women's dresses: hemline, neck style, pattern, silhouette and sleeve type. Table 2 shows the number of images with human tags and the number of classes for each attribute (Class details are shown in Figure 1).

Table 2. Dress visual attributes classification datasets

	hemline	neck style	pattern	silhouette	sleeve type
Number of classes	5	11	10	5	5
Number of images	76,921	75,439	64,825	192,619	77,352

3.2 Performance evaluation

For each type of attribute, we randomly split the corresponding dataset into 80% for training and 20% for testing. As our attribute classification problems have multiple classes, we picked the prediction class with maximum probability from the model. To evaluate the performance of the trained models, we used a multi-class accuracy metric (definition: number of images being correctly predicted to the classes/ total number of images). This allowed us to evaluate the overall model performance, the per-class accuracy, and the blended average class accuracy.

Four base models have been tested for the refine model training. ImageNet, Softlines_2M, Softlines_Top500 and Softlines_Top2000. Experiments show that Softlines_2M is only comparable or slightly better (within about 1%) than the ImageNet base model, so we only present ImageNet base model results here. Table 3 shows the results.

Table 3. Dress visual attributes classifier performance based on different model settings

	ImageNet		Softlines_Top500		Softlines_Top2000	
	multi-class accuracy	average per-class accuracy	multi-class accuracy	average per-class accuracy	multi-class accuracy	average per-class accuracy
hemline	85.05%	75.47%	86.98%	75.27%	87.21%	75.99%
neck_style	71.43%	57.07%	76.07%	58.49%	77.26%	60.01%
pattern	74.04%	42.66%	80.43%	51.30%	81.75%	57.91%
silhouette	84.91%	83.37%	86.30%	85.34%	86.81%	85.57%
sleeve_type	87.76%	82.25%	89.75%	86.26%	90.31%	87.14%

From the results, we can see that Softlines_Top500 and Softlines_Top2000 perform noticeably better than those based on Softlines_2M. This is likely due to the fact that some of the 8,687 labels (e.g., those have very low frequencies) used by Softlines_2M are not meaningful at all, which only mislead the training of the deep learning models.

Table 4. Accuracy comparison using cross validation data vs. production data

	multi-class accuracy in cross validation data	multi-class accuracy in production data
hemline	87.21%	73.70%
neck_style	77.26%	66.70%
pattern	81.75%	86.2%
silhouette	86.81%	82.50%
sleeve_type	90.31%	87.60%

3.3 Model evaluation in production

All the accuracy metrics reported are the result of cross-validation. We had to generate real validation data accuracy in production. To achieve this, we first productionized the best performing model, Softline_Top2000. We then collected 3-month data in production to evaluate the model performance. Table 4 shows the multi-class accuracy in cross-validation data versus production data. We can see that the accuracy dropped in production for all models except the pattern model. The potential reasons for accuracy decreases could be: 1) The cross-validation data may have contained similar images to the training data from one parent ASIN, which made the model originally perform well, 2) the training data didn't cover the variance of production data, 3) New styles came in as trends evolve. These new styles could be a hybrid of multiple classes which would make the model hard to distinguish.

4 Results analysis

4.1 Neck-style model with low accuracy

In the test dataset, "Neck_style" had the lowest performance (only achieved 77.26%) while other classifiers such as "hemline", "silhouette", "sleeve_type" and "pattern" all achieved 80%+. This is because "neck_style" only uses a small area of the whole dress image, but we use the whole image as an input. All the other image features outside of the neck area become noise signals which makes it difficult for the classifier.

4.2 Data imbalance issue

It is worth mentioning that the difference between averaged per-class accuracy and multi-class accuracy is much larger for "hemline" and "neck_style" compared to "silhouette" and "sleeve_type". The differences are most likely due to the fact that "hemline" and "neck_style" have class imbalance issues in their datasets (Figure 5). The CNN model training can be highly biased towards the majority classes where high multi-class accuracy can be obtained. However, for the attribute classes that have much fewer training images, we cannot achieve good per-class classification performance. This makes the average per-class accuracy much lower (Table 5).

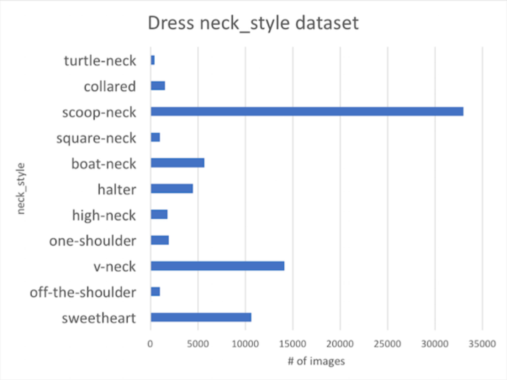


Figure 5. Example of imbalanced data distribution in neck style

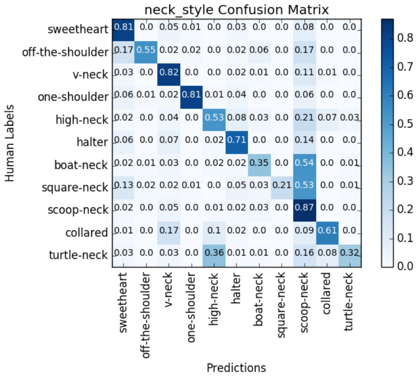


Figure 6. Example of confusion matrix for neck_style classifier

4.3 Labeling issue

In Figure 6, we can see that some attribute classes are easily confused with each other ("square_neck" vs. "scoop_neck" under "neck_style"). These attribute classes are not easily visually distinguishable, so it can be quite difficult to tell which attribute classes they should belong to. We are exploring re-designing our labeling to combine classes or potentially support multi-labeling.

Table 5. Example of different accuracy for imbalanced classes within the Neck Style classifier.

class	Sweet heart	off-the-shoulder	v-neck	one-shoulder	high-neck	halter
accuracy	81.33%	55.33%	81.61%	81.09%	52.51%	70.95%
class	**boat-neck**	**square-neck**	**scoop-neck**	**collared**	**turtle-neck**	**mean**
accuracy	35.37%	21.21%	86.85%	61.36%	32.47%	60.01%

4.4 Size of training data

With more human labeled data available, we increased the size of our training dataset and reran the training process with the same architecture. Results are shown below (Table 6). Interestingly, we noticed that accuracy did not increase for most of the models except hemline. The potential reason could be the huge variance within the training images as displayed in Figure 2. Purely increasing the amount of training images would not benefit us significantly. Any useful signal would be cancelled out by more noisy signals.

number of image/ cross validation accuracy	**hemline**	**neck_style**	**pattern**	**silhouette**	**sleeve_type**
New	270,109 91.2%	204,805 73.9%	64,108 78.17%	380,368 85.6%	264,795 89.9%
Old	76,921/ 87.21%	75,439/ 77.26%	64,825/ 81.75%	192,619/ 86.81%	77,352/ 90.31%

Table 6. Performance with different amounts of training data

5 Summary and Future work

In summary, we demonstrated the success of using a convolutional neural network to classify visual attributes for Softlines ASINs. Overall, the model achieved 80% accuracy. The classified attributes are currently being used for browse node classification and a visual search feature for Dresses. We are continuing to improve model accuracy and catalog coverage so that we can integrate with the search index and improve the quality of search results for customers.

We will be testing several different methods to improve accuracy and coverage: 1) The data imbalance problem exists for almost all attributes (especially for "hemline", "neck_style" and "pattern"). We can assign a pre-defined weight of each training instance to balance the attribute classes. Work needs to be done within MXNet for passing the pre-defined weights into the CNN model during the training phase. 2) For attribute types (e.g., "neck_style") that target a small part of a whole image, we can introduce some heuristic image detector to pre-identify each area of interest. 3) For the labeling issue, we can redefine our classification labels to make them technically more classifiable while focusing first on what the customer is looking for. 4) We also need to find tune the parameters of MXNet (e.g. batch size, learning rate, and weight decay etc.) to improve model performance. 5) Currently, we only use image data as features, however, some attribute information is easier to extract from text (e.g. title, description etc.) than image. We plan on building a deep learning model that combines image and text features together. We will need to redesign the architecture of our current CNN to adapt to this method.

From a business perspective, our goal is to use our models to directly improve the customer experience at Softlines. To that end, we are setting up Weblabs to evaluate the business impact on a variety features. This includes testing whether model-generated attributes improve browse conversion and whether cleaner keyword data improves the quality of search results. We also plan on expanding to additional product categories and international marketplaces. We will continue to prioritize our classification models based on attributes that are most important to the fashion customer. This includes supporting a variety of occasions, style aesthetics, as well as fashion trends. As we increase the amount of relevant metadata in our catalog, we will be able to power more customer discovery experiences, such as automatically clustering products into personalized collections, adding visual-based similarities to the detail page, helping customers shop by occasion, and so on.

Acknowledgments

Thanks to fashion specialists from Softlines for collecting the images used in this paper.

References

[1] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[2] Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. CoRR, abs/1409.4842, 2014.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, november 1998.

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

[6] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed System, In NIPS Workshop on Machine Learning Systems (LearningSys), 2016