Convolutional Neural Networks for Fashion Classification and Object Detection

Brian Lao
bjlao@stanford.edu

Karthik Jagadeesh
kjag@stanford.edu

Abstract

Fashion classification encompasses the identification of clothing items in an image. The field has applications in social media, e-commerce, and criminal law. In our work, we focus on four tasks within the fashion classification umbrella: (1) multiclass classification of clothing type; (2) clothing attribute classification; (3) clothing retrieval of nearest neighbors; and (4) clothing object detection. We report accuracy measurements for clothing style classification (50.2%) and clothing attribute classification (74.5%) that outperform baselines in the literature for the associated datasets. We additionally report promising qualitative results for our clothing retrieval and clothing object detection tasks.

1. Introduction

Clothing in many cultures reflects characteristics such as age, social status, lifestyle and gender. Apparel is also an important descriptor in identifying humans, e.g. ”the man wearing an orange jacket” or ”the woman in red high heels.” Given the role of clothing apparel in society, ”fashion classification” has many applications. For example, predicting the clothing details in an unlabeled image can facilitate the discovery of the most similar fashion items [1] in an e-commerce database. Similarly, classification of a user's favorited fashion images can drive an automated fashion stylist, which would provide outfit recommendations based on the predicted style of a user. Real-time clothing recognition can be useful in the surveillance context [2], where information about individuals' clothes can be used to identify crime suspects. Fashion classification also facilitates the automatic annotation of images with tags or descriptions related to clothing, allowing for improved information retrieval in settings such as social network users' photos.

Depending on the particular application of fashion classification, the most relevant problems to solve will differ. We will focus on optimizing fashion classification for the purposes of annotating images and discovering the most similar fashion items to a fashion item in a query image.

Some of the challenges for this task include: classes of clothing can share similar characteristics (e.g. the bottoms of dresses vs. the bottoms of skirts), clothing can easily deform due to their material, certain types of clothing can be small, and clothing types can look very different depending on aspect ratio and angle.

2. Problem Statement

Our problem (Figure 1) is defined as follows: given a query image that contains clothing, (a) predict the clothing type through multi-class classification (clothing type classification), (b) predict the clothing attributes through attribute classification (clothing attribute classification), (c) find the most similar pieces of clothing in the dataset (clothing retrieval), and (d) determine a set of regions within an image that contain clothing objects.

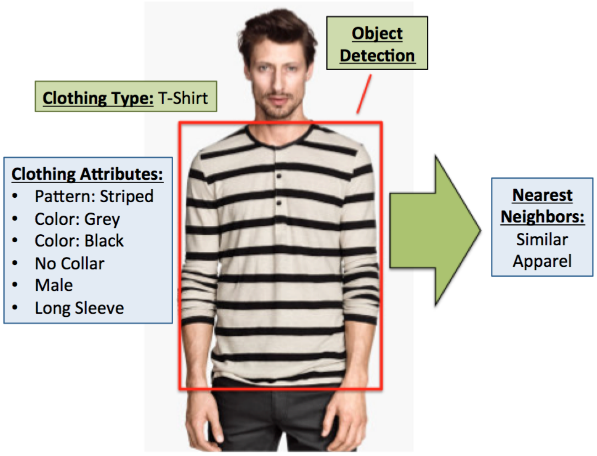
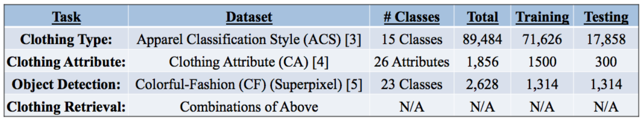


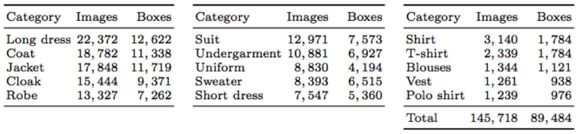
Figure 1. Summary of our four fashion classification tasks: given a test image that contains clothing, detect the clothing items in the image, classify by clothing type and attributes, and retrieve the similar clothing items.

Clothing type classification, clothing attribute classification, clothing detection, and clothing retrieval can be viewed as separate sub-problems within the umbrella of fashion classification. We plan to approach each sub-problem with convolutional neural networks (CNNs). CNNs have rarely been applied to the fashion domain. Recently, Hara et al. (2014) [7] adapted Girshick's Region-



Task	Dataset	# Classes	Total	Training	Testing
Clothing Type:	Apparel Classification Style (ACS) [3]	15 Classes	89,484	71,626	17,858
Clothing Attribute:	Clothing Attribute (CA) [4]	26 Attributes	1,856	1500	300
Object Detection:	Colorful-Fashion (CF) (Superpixel) [5]	23 Classes	2,628	1,314	1,314
Clothing Retrieval:	Combinations of Above	N/A	N/A	N/A	N/A

Figure 2. Summary of the datasets we use for the four fashion classification tasks



Category	Images	Boxes	Category	Images	Boxes	Category	Images	Boxes
Long dress	22,372	12,622	Suit	12,971	7,573	Shirt	3,140	1,784
Coat	18,782	11,338	Undergarment	10,881	6,927	T-shirt	2,339	1,784
Jacket	17,848	11,719	Uniform	8,830	4,194	Blouses	1,344	1,121
Cloak	15,444	9,371	Sweater	8,393	6,515	Vest	1,261	938
Robe	13,327	7,262	Short dress	7,547	5,360	Polo shirt	1,239	976
						Total	145,718	89,484

Figure 3. 15 classes and # of images or bounding box cropped images for the Apparel Classification with Style (ACS) dataset [3]

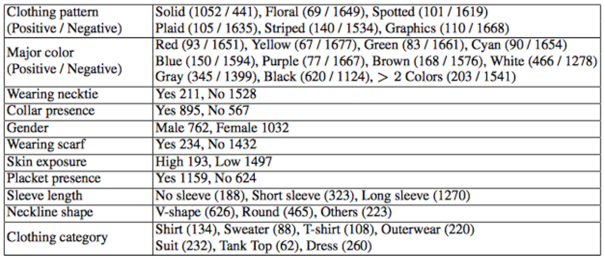
CNN (R-CNN) object detection model [8] to detecting fashion items worn by an individual. Hara applied the R-CNN model to an unreleased modified Fashionista Dataset [9], which contains 685 images annotated with bounding boxes after conversion from pixel-label annotations. Fashion classification has more generally consisted of non-CNN approaches, and we will discuss relevant related work throughout the paper in the appropriate sections. Experimentally, we will focus on applying CNNs to classification tasks that best facilitate image annotation and finding the most similar clothing to a query item, as well as building on top of the R-CNN model to identify clothing objects in the image.

A summary of the datasets that we use for our own fashion classification tasks can be found in Figure 2.

2.1. Clothing Type Classification

Clothing type classification is the multiclass classification problem of predicting a single label that describes the type of clothing within an image. Thus, clothing type datasets will include images of clothing annotated with a label such as hat, jacket, or shoe. We will be using the Apparel Classification with Style (ACS) Dataset [3], which contains 89,484 images that been cropped based on bounding boxes aimed at encapsulating the clothing on an individual's upper body. Each image is labeled with one of 15 hand-picked clothing categories (Figure 3).

Bossard et al. (2012) [3] used the ACS dataset to extract features including Histogram of Oriented Gradients (HOG), Speeded Up Robust Features (SURF), Local Binary Patterns (LBP), and color information. Bossard then used these features to perform multiclass classification with One vs. All SVM, random forests, and transfer forests, achieving average accuracies of 35.03%, 38.29% and 41.36%, respectively. Using our CNN, we exceed these accuracy baselines on the ACS dataset.



Clothing pattern (Positive / Negative)	Solid (1052 / 441), Floral (69 / 1649), Spotted (101 / 1619) Plaid (105 / 1635), Striped (140 / 1534), Graphics (110 / 1668)
Major color (Positive / Negative)	Red (93 / 1651), Yellow (67 / 1677), Green (83 / 1661), Cyan (90 / 1654) Blue (150 / 1594), Purple (77 / 1667), Brown (168 / 1576), White (466 / 1278) Gray (345 / 1399), Black (620 / 1124), > 2 Colors (203 / 1541)
Wearing necktie	Yes 211, No 1528
Collar presence	Yes 895, No 567
Gender	Male 762, Female 1032
Wearing scarf	Yes 234, No 1432
Skin exposure	High 193, Low 1497
Placket presence	Yes 1159, No 624
Sleeve length	No sleeve (188), Short sleeve (323), Long sleeve (1270)
Neckline shape	V-shape (626), Round (465), Others (223)
Clothing category	Shirt (134), Sweater (88), T-shirt (108), Outerwear (220) Suit (232), Tank Top (62), Dress (260)

Figure 4. For each of the 26 classes in the Clothing Attribute dataset, this table characterizes the # of images with that label. [4]

2.2. Clothing Attribute Classification

Clothing attribute classification is the problem of assigning attributes such as color or pattern to an article of clothing. The attributes will be contained within a binary vector of possessing or not possessing certain attributes within this selection of attributes. Whereas a fashion item will only have a single clothing type such as "jacket," the item may have multiple clothing attributes. We will be using the Clothing Attribute (CA) Dataset [4], which contains 1856 upper-body clothing images annotated from a pool of 26 attributes (Figure 4). An example image from the data set might have attributes such as: no necktie, has collar, men's, solid pattern, blue, white.

Navarro et al. (2014) [6] extracted LBP and HOG features from the CA dataset. Applying SVM and Random Forest classifiers to the pattern attributes, Navarro achieved 78.44% and 81.76%, respectively. On the color attributes, 76.71% and 82.29% accuracies were achieved, and similar accuracies were achieved for the sleeve length, collar, and necktie attributes. We will use these accuracy measurements as a baseline to determine the effectiveness of the multi-label CNN architecture we are using.

2.3. Clothing Retrieval

Clothing retrieval encompasses the task of finding the most similar clothing items to a query clothing item. We hypothesize that the nearest neighbors will more similarly match the query image when incorporating features learned when using both (a) the clothing type dataset and (b) the clothing attribute dataset. Although there will likely be some overlap between the features learned for the two datasets, we also predict that each dataset will have its own set of unique features and weightings, so combining the data will result in a more robust set of weights.

The ACS and CA datasets are focused on upper-body clothing, but we hope that our CNN models can be generalized to clothing for other parts of the body as well.

	face	*sunglass*	*hat*	*scarf*	*hair*
HEAD	3629 (1337)	292 (220)	782 (190)	965 (116)	10806 (1330)
	3675 (1341)	272 (205)	620 (160)	890 (91)	10732 (1332)
	blazer	*T-shirt*	*blouse*	coat	*sweater*
UPPER	3811 (178)	6172 (460)	8669 (474)	3999 (170)	3030 (133)
	3917 (176)	6068 (457)	8198 (461)	4201 (186)	2795 (115)
	jeans	*legging*	*pants*	*shorts*	*skirt*
LOWER	2860 (113)	3124 (123)	3418 (103)	2451 (226)	10214 (438)
	2388 (87)	2699 (110)	2825 (86)	3021 (276)	10353 (445)
	shoe	*socks*	*stocking*		
FOOT	5184 (1300)	242 (83)	1831 (131)		
	5287 (1301)	284 (87)	2316 (157)		
	skin	*belt*	*bag*	*dress*	*bk*
OTHER	26830 (1324)	1056 (390)	6301 (723)	11300 (342)	452003 (1341)
	26690 (1330)	1174 (432)	6692 (739)	11680 (336)	452338 (1341)

Figure 5. For each category in the Colorful-Fashion dataset, the number of superpixel patches for the training and testing subsets are shown in the first and second rows, respectively. The number of images containing the category is shown in parenthesis. [5]

2.4. Clothing Object Detection

Clothing Object Detection consists of detecting the specific regions of the clothing objects present in a given image. For example, given an image of an individual wearing a full outfit, clothing object detection involves the prediction of bounding boxes that would capture the distinct articles of clothing such as the shirt, pants, and shoes.

We used the Colorful-Fashion (CF) (Figure 5) dataset for the clothing object detection task. Our approach of fine-tuning an R-CNN model [8] requires bounding box annotations, and the CF dataset is superpixel-labeled. Thus, using the edges of the superpixel-labeling, we converted the labels into ground-truth bounding box images.

To increase the number of training patches, we ran Selective Search (with parameter setting of 0.5 intersection over union (IOU)) on the training images. Running Selective Search on each image resulted in 2500 windows per image, with 2,895,511 window proposals generated for the training set and 2,691,491 window proposals generated for the test set. For every bounding-box proposal, we calculate the IoU against each ground-truth bounding box. If a bounding-box proposal has IoU larger than 0.5 for a ground-truth bounding box, we use the proposal as another training sample for that class. When a bounding-box proposal has IoU less than 0.1 for all ground-truth bounding boxes, we use it as a training sample for the background class. For each stochastic gradient descent iteration, 75% of the batch is sampled from background windows while 25% is sampled from foreground windows (offsetting the bias of otherwise a much greater proportion of background windows).

3. Technical Approach and Models

3.1. Clothing Type Classification

We assessed accuracy, precision, recall, and F1-Score on the ACS dataset using the standard AlexNet Convolutional network which had been pretrained using ImageNet. Although ImageNet contains many classes unrelated to clothing or humans, features extracted from pre-trained ImageNet models have proven useful even for predicting on dissimilar datasets [11]. Given that our ACS dataset contains 89,484 images - a decent sized dataset - we hypothesized that starting our fine-tuning at earlier CaffeNet layers would optimize performance. We conducted our fine-tuning through a two-phase process in accordance with Branson et al [10]. In the first phase, we substitute a new inner product layer (output: 15 for the 15 classes) with Softmax classifier. After hyperparameter tuning (LR: 3e-4), we train for 25,000+ iterations. In the second phase, we unfreeze all layers. After hyperparameter tuning (LR: 6e-5), we train for 10,000+ iterations.

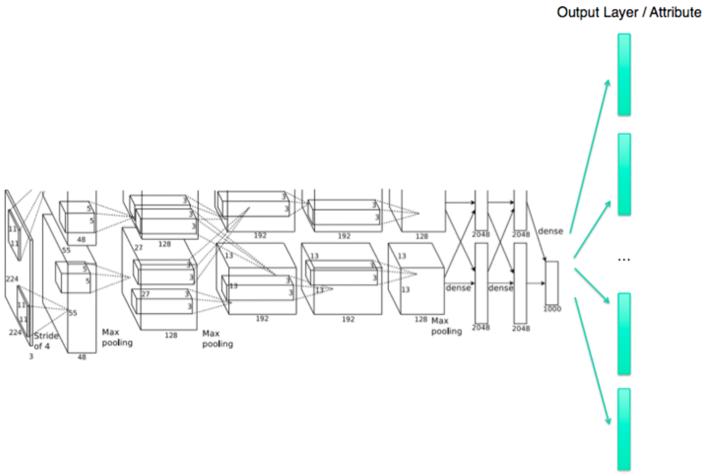


Figure 6. Standard AlexNet architecture, but with 26 output layers. One for each attribute that we are predicting.

3.2. Clothing Attribute Classification

We have a total of 26 attributes such as color, sleeve length, neckline, etc which have been labeled in our dataset. For this task, we will develop a model for multi-label based classification on a set of 1,700 labeled images. We are building on the original CaffeNet architecture, but have replaced the output Softmax layer with 26 Softmax layers (1 per attribute) that each take as input the activations from the Fully Connected Layer 7 (fc-7). Each of these output layers will be randomly initialized and retrained independently using the labeled data for a specific attribute.

Our CA dataset for attribute classification has significantly fewer images (1856) compared to the ACS dataset (89,484) for type classification. With a smaller dataset to fine-tune upon, we expect that performance might be optimized if we fine-tune starts at a later-stage network layer.

3.3. Clothing Retrieval

One approach to solving this problem is to use a Nearest Neighbors approach on the raw pixels in the image. This will be a very limiting strategy because it will do a pixel by pixel comparison to find images that match on a per pixel basis. Instead, we look at the activations from the 7th layer of the fine-tuned CaffeNet convolutional network (fc-7), and used these context based features to represent each image. Each feature in this setting represents a high level aspect of the input pixel features which is not as captured by a single pixel in the original method.

Given that we are finding the nearest neighbors based on the "code" layer of our CNN, we expect that the nearest neighbors to a query image will sometimes be distant in the pixel space.

3.4. Clothing Object Detection

We use the Caffe-implemented R-CNN pre-trained model (trained on ILSCVRC13) on the CF dataset modified with bounding box labels and extra training patches. Following the Branson et al. tuning procedure, we perform our first phase of parameter tuning by substituting a new inner product layer (output: 23, for 22 classes + 1 background class) with Softmax classifier. After hyperparameter tuning (LR: 0.001), we train for 20,000+ iterations. In the second phase of parameter tuning, we unfreeze all layers in the network and train for 10,000+ iterations. We use the snapshotted models with the best validation accuracies.

4. Results

4.1. Clothing Type Classification

Our results are summarized in Figure 7. For our modified CaffeNet model trained with frozen layers except a new inner product layer, we achieve a 46.0% accuracy on the test set. Regarding our modified CaffeNet model fine-tuned on all layers, we achieve a 50.2% accuracy on the test set. As expected, fine-tuning on all layers non-trivially improved our accuracy. Both our CNN models outperform the SVM, random forests, and transfer forests models of Bossard et al., the paper from which the ACS fashion dataset originated. Precision, recall, and F1 scores for each class is shown in Figure 8.

4.2. Clothing Attribute Classification

Attribute classification proved to be a very challenging task for certain attributes, but much more successful in classifying other attributes. We see measurements greatly varying from close to 100% accuracy for attributes like brown or yellow, to as low as 20% for attributes like placket and solid.

We have computed the accuracy measurements a category by category basis as well as looked at the average accuracy across all labels for each iteration. These statistics are shown in the following 2 figures (Figure 9, Figure 10).

Model	**Accuracy**
SVM (Bossard et al.)	35.0%
Random Forests (Bossard et. Al)	38.3%
Transfer Forests (Bossard et. Al)	41.4%
Fine-tuned Fully-Connected Layers CaffeNet	46.0%
Fine-tune All Layers CaffeNet	50.2%

Figure 7. Comparison of models for the clothing type classification task. We see that our CNN models outperform the baseline models in Bossard et al. for the ACS dataset.

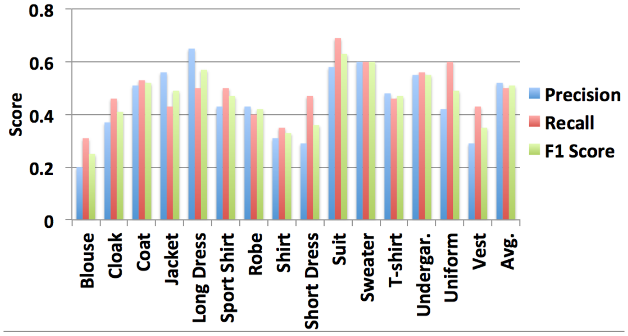


Figure 8. Our CaffeNet model fine-tuned on all layers using the ACS dataset: precision, recall, and F1 scores on a per-class basis.

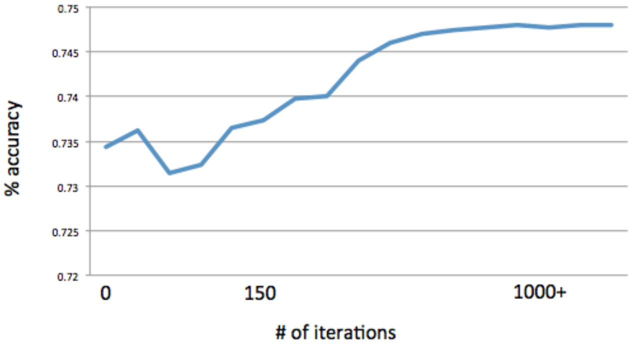


Figure 9. As we run the model for more and more iterations we start to see that the average accuracy is stabilizing around 74.5%.

4.3. Clothing Retrieval

We pulled the set of features computed from the second fully connected layer in AlexNet (fc-7). To test the effectiveness of using these features we looked at 5 query images and the 6 Nearest Neighbors that our method is able to pick up for each query. The results from this experiment are shown in Figure 9.

For each query image, the KNN clothing retrieval algorithm is able to pull up images that are similar to the query beyond just looking at a pixel by pixel comparison.

As a general method of quantitatively testing the effi-

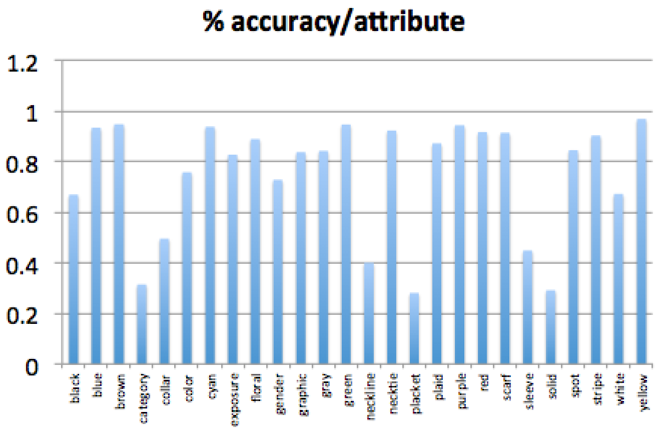


Figure 10. This table shows the % accuracy we have per category when predicting attributes on the test data.

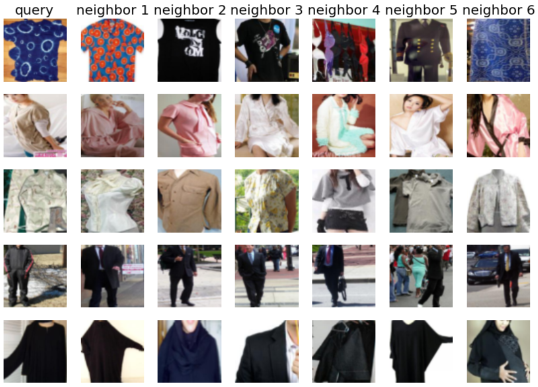


Figure 11. 5 query images with 6 Nearest neighbor images retrieved for each query image. We see clothes similar to the query image being displayed.

ciency of the methods and whether the images we are retrieving are relevant, we looked at how often we can correctly classify the image into the correct clothing type category. The KNN algorithm using activations from fc-7 is able to correctly predict the class of the image 40.2% of time. This accuracy is very close to the accuracy level that the transfer forest methods used in previous papers was able to reach. Given the limitations of KNN we see these results as big reinforcement that the model is learning highly relevant features for clothing classification.

4.4. Clothing Object Detection

During phase-one training of the R-CNN model substituted with a new inner product layer, we achieve a 91.25% top validation accuracy (validation batches of 25% foreground windows and 75% background windows). During phase-two training of the R-CNN model with all layers unfrozen, we achieve a 93.4% top validation accuracy. An example of running our top model on a test image is shown in Figure 12, where the top 7 detection classes for clothing items are shown. For illustrating how our object detector model performs within certain classes, we provide Figure 13. The figures illustrates the top 3 scoring windows for the ”shoe” class and for the ”belt” class, respectively, after applying greedy non-maximum suppression with an overlap threshold of 0.3. To assess our object detection models, our future work will include calculations of mean average precision (mAP), a common metric to evaluate the quality of object detectors.

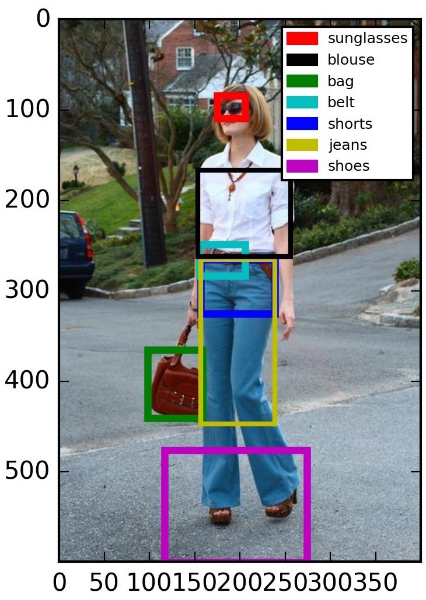


Figure 12. Clothing Object Detection: bounding boxes for top 7 scoring clothing classes.

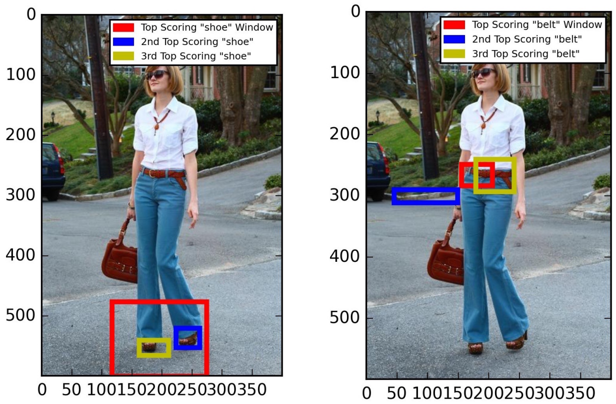


Figure 13. Clothing Object Detection: Left: bounding boxes for the windows with the top 3 scores for ’shoe’ class. Right: bounding boxes for the windows with the top 3 scores for ’belt’ class. Greedy non-maximum suppression is performed with an overlap of 0.3 in order to reduce overlapping windows

5. Discussion

5.1. Clothing Type Classification

With our top model, we achieved a 50.2% accuracy, which outperforms the Bossard et al. Transfer Forests baseline of 41.4%. As can be seen in Fig. 8, the ”Blouse” and ”Shirt” classes scored lowest on F1 score. Intuitively, this is likely due to the similarity in visual characteristics between a blouse and a shirt, leading to classification difficulty in distinguishing the two classes. ”Shirt” (not ”T-shirt”) encompasses articles of clothing that are often defined by buttons and collars, just like many blouses. Figure 14 illustrates a blouse image and a shirt image taken from the ACS dataset. Given the non-traditionality of many fashion items, there is likely some subjectivity involved in manually labeling images when categories such as blouse and shirt have many overlapping characteristics. ”Suit” and ”Sweater” classes performed best. This is expected, given a suit’s distinct clothing characteristics(often darker color, uniform blazer structure, long-sleeve) and a sweater’s distinct clothing characteristics (often wool-like material, long-sleeve).

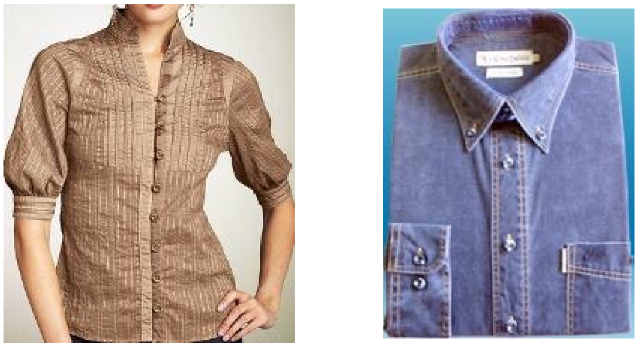


Figure 14. Images taken from the ACS dataset: Left: image from ”blouse” category.” Right: image from ”shirt” category.

5.2. Clothing Attribute Classification

We were able to see a marked improvement in attribute classification over existing clothing attribute classifiers. The best average accuracy across all category types found in previous research we looked at was 70% overall accuracy[5]. While our method performs with an overall average accuracy of 74.5%.

As seen in figure 10 the accuracy for a specific attribute varies greatly between depending on the type of attribute. In general it does very well at classifying the color attributes, but much worse when looking at subtle attributes like pattern types in the clothes. Looking deeper in the training data we noticed that many of the images in categories such as placket are tough to properly distinguish even using the human eye, and there are not enough samples in our data to learn a clear pattern.

It would great help the model learn if we created a larger dataset with more training examples for many of these attributes that are currently causing confusion for our classifier.

5.3. Clothing Retrieval

Clothing Retrieval using the fc-7 activations performs significantly better than the pixel based Nearest Neighbors approach. Based on the images retrieved in the previous section for each query image we can see the clothing retrieval task recognizes colors, clothing styles, and patterns in clothes.

Colors - The colors found in the query image are consistently found in the results that are retrieved by the KNN algorithm. An image with a black suit clusters closely with other images of suits, even if it is shifted to be in a completely different part of the image.

Texture - Similarly the pink and silk blouse clusters with clothes that have light shades such as white and light blue, but all the images show a shiny texture that resembles the query image.

5.4. Clothing Object Detection

The bounding boxes and labels for the test image (Fig. 11) are promising. The individual in the test image is wearing 6 articles of clothing (sunglasses, blouse, belt, jeans, bag, and shoes). The model mislabeled ”shorts,” likely due to the visual characteristics of the blue bounding box resembling shorts (hip-length, jean-like material). However, the remaining top 6 predicted classes predicted all match the ground-truth 6 articles of clothing. Surprisingly, our model was able to correctly detect the woman’s ”blouse,” which is a class that we saw performed poorly in the clothing type classification task. In the top scoring windows for ’shoe’ (Fig. 13), we were impressed that the windows are accurate despite the individual shoes in the image being partially covered by the woman’s jeans. In the top scoring windows for ’belt’ (Fig. 13), we see that the 2nd top window identified the sidewalk curb as a ’belt,’ indicating the classifier’s vulnerabilities in labeling long, thin, horizontal non-belt objects as belts.

6. Conclusions and Future Work

We plan on continuing our fashion classification work beyond the CS231N course.

6.1. Alternative Architectures

For this project, we built on top of the default AlexNet architecture using the Imagenet pre-trained weights as a starting point for our work. As a next step, we will try modified architectures more tailored for practical fashion classification applications. For example, we are in the process of implementing spatial pyramid pooling layers, which have been shown to speed up the R-CNN method by 24-102x. [14]

6.2. Data augmentation

In our fashion classification tasks, we experimented with data augmentation techniques such as mean subtraction, cropping, and resizing. We plan on employing further data augmentation techniques discussed in class, such as rotating the image and flipping along the horizontal axis to further perturb the input image.

6.3. Model Evaluation and Transfer

In our experiments, we perform classification tasks using three different clothing datasets. We apply transfer learning from models pre-trained primarily on ImageNet. However, we plan to assess model accuracies after applying transfer learning between our own models that are now trained on fashion datasets. Regarding our object detection models, it is difficult to assess the model's efficacy based only on validation accuracy. Future steps will thus include assessment of mean average precision (mAP) for the test set.

Acknowledgements

All the team members contributed equally to all aspects of the project.

References

[1] Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., Yan, S.: Street-to-Shop: Cross-Scenario Clothing Retrieval via Parts Alignment and Auxiliary Set. *CVPR* (2012).

[2] Yang, M., Yu, K.: Real-time clothing recognition in surveillance videos. In: *18th IEEE International Conference on Image Processing* (2011).

[3] Bossard, Lukas, et al. ”Apparel classification with style.” *Computer VisionACCV 2012. Springer Berlin Heidelberg, 2013. 321-335.*

[4] Chen, Huizhong, Andrew Gallagher, and Bernd Girod. ”Describing clothing by semantic attributes.” *Computer VisionECCV 2012. Springer Berlin Heidelberg, 2012. 609-623.*

[5] Liu, Si, et al. ”Fashion parsing with weak color-category labels.” *Multimedia, IEEE Transactions on 16.1 (2014): 253-265.*

[6] Lorenzo-Navarro, Javier, et al. ”Evaluation of LBP and HOG Descriptors for Clothing Attribute Description.” Video Analytics for Audience Measurement. *Springer International Publishing, 2014. 53-65.*

[7] Hara, Kota, Vignesh Jagadeesh, and Robinson Piramuthu. ”Fashion Apparel Detection: The Role of Deep Convolutional Neural Network and Pose-dependent Priors.” arXiv preprint *arXiv:1411.5319* (2014).

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014.

[9] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. *CVPR*, 2012.

[10] Branson, Steve, et al. ”Bird species categorization using pose normalized deep convolutional nets.” *arXiv preprint arXiv:1406.2952 (2014).*

[11] Y.Jia,E.Shelhamer,J.Donahue,S.Karayev,J.Long,R.Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv,* June 2014.

[12] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” *Advances in neural information processing systems* 2012.

[13] Razavian, Ali Sharif, et al. ”CNN Features off-the-shelf: an Astounding Baseline for Recognition.” *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE, 2014.*

[14] He, Kaiming, et al. ”Spatial pyramid pooling in deep convolutional networks for visual recognition.” *arXiv preprint arXiv:1406.4729 (2014).*