Fashion Style in 128 Floats: Joint Ranking and Classification using Weak Data for Feature Extraction

Edgar Simo-Serra and Hiroshi Ishikawa
Department of Computer Science and Engineering
Waseda University, Tokyo, Japan
esimo@aoni.waseda.jp hfs@waseda.jp

Abstract

We propose a novel approach for learning features from weakly-supervised data by joint ranking and classification. In order to exploit data with weak labels, we jointly train a feature extraction network with a ranking loss and a classification network with a cross-entropy loss. We obtain high-quality compact discriminative features with few parameters, learned on relatively small datasets without additional annotations. This enables us to tackle tasks with specialized images not very similar to the more generic ones in existing fully-supervised datasets. We show that the resulting features in combination with a linear classifier surpass the state-of-the-art on the Hipster Wars dataset despite using features only 0.3% of the size. Our proposed features significantly outperform those obtained from networks trained on ImageNet, despite being 32 times smaller (128 single-precision floats), trained on noisy and weakly-labeled data, and using only 1.5% of the number of parameters.[1].

1. Introduction

With the emergence of large-scale datasets and the appearance of deep networks with millions of parameters, researchers have started to replace hand-crafted global image features such as GIST [21] with those obtained from intermediate representations of deep networks trained for classification on large datasets [44]. Although this has led to a great improvement over the previous generation of features, these networks are learned in a fully-supervised manner on large amounts of data with very costly and time-consuming annotation. Features learned on one dataset can be used on another, but naturally not all datasets are equal [32], and thus features taken from networks trained on ImageNet [7] will not work as well on datasets with very different visual characteristics, such as the scene classification dataset Places [49], and vice versa. While unsupervised feature learning exists as an alternative to supervised learning, the lack of guidance to what to learn given by explicit labels makes it a much more complex task [19].

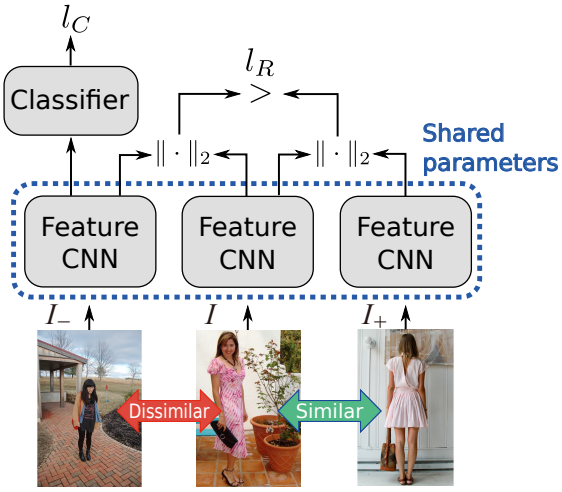

Figure 1: Overview of the proposed feature learning approach. We train a feature extraction on weakly annotated data by jointly training a feature extractor network with a classification network. For training, an anchor image (center) is provided in conjunction with a similar image (right) and a dissimilar image (left) according to a metric provided on the weak noisy annotations. The classification loss l_C serves to learn useful feature maps while the ranking loss l_R on the triplet of Feature CNN encourages them to learn a discriminative feature representation.

However, images obtained from the Internet usually have associated metadata which, although often inaccurate, can be used as *weak* labels. In this paper, we study how to exploit data with weak labels in order to obtain high-quality compact discriminative features without additional annotations. With such features, we tackle tasks in which the images are more specific and not very similar to those of existing fully-supervised datasets such as ImageNet or Places.

In this work, we focus on the domain of fashion images, which have only recently become the focus of research [40]. These images have several characteristics that make them

[1] Models available at http://hi.cs.waseda.ac.jp/ esimo/research/stylenet/

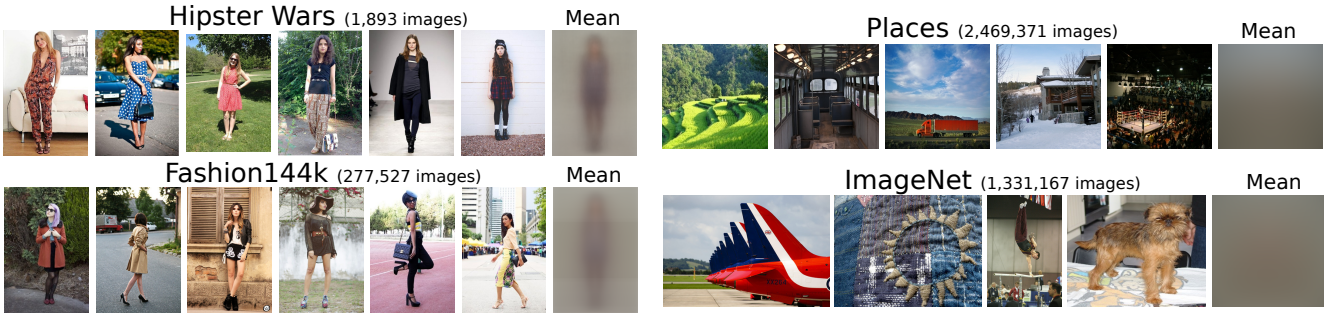

Figure 2: We show example images and the mean image from the Hipster Wars [16], Fashion144k [27], Places [49], and ImageNet [7] datasets. In both fashion-related datasets we can make out a human silhouette, although it is significantly more diffuse in the Fashion144k dataset due to the much larger pose variation. In the Places dataset mean image we can see a gradient where the top of the image is clearer, likely corresponding to the sky in many images. While in the ImageNet mean we see a much more uniform image with a slightly clearer area in the center of the image. Unlike the other datasets, Fashion144k only has weak labels. With our approach we are able to exploit the Fashion144k for training to evaluate on the Hipster Wars dataset, outperforming fully supervised approaches that use larger datasets such as ImageNet or Places.

very challenging to tackle with computer vision. On one hand, they have small local details such as accessories that only depend on a very small part of the image, making segmentation very challenging [26, 40, 43]. On the other hand, we still have to consider more global properties such as the fashion style [16, 27, 34], which depend jointly on the various items in the image. These difficulties, along with the fact that the images generally have a 3:4 aspect ratio and tend to have much brighter colors, cause the features taken from networks trained on ImageNet or Places to generalize poorly to fashion-oriented tasks. While no large fully-annotated fashion dataset exists, there are many datasets, such as the Paperdoll [41] and Fashion144k [27] datasets, that have a large amount of weak labels exploitable for learning. In this work, we take advantage of these noisy labels to learn compact discriminative features with deep networks that can then be used in other challenging fashion-related tasks, such as style classification [16], in which they greatly outperform the state-of-the-art and other pretrained CNN baselines, while having 1.5% the number of parameters and being the size of a SIFT descriptor [20]: 128 floats and 32$\times$ smaller than the best competing approach.

Instead of training networks for classification and using an intermediate-layer representation as a feature vector, we propose performing joint classification and ranking as shown in Fig. 1. The ranking acts as a soft constraint on the intermediate layer and encourages the model to learn more representative features guided by the classifier. We perform ranking by considering three images simultaneously: we first pick an anchor image and then pick an image that is similar to the anchor and one that is different. We establish a similarity metric by exploiting the weak user-provided labels, which are also used as a classification target. By simultaneously considering both ranking and classification, we are able to outperform approaches that use either ranking or classification alone. Our approach allows us to efficiently make use of weak labels to learn features on datasets closer to the target application, as shown in Fig. 2.

In summary, our novel approach for feature learning:

- Exploits large amounts of *weakly-labeled* data commonly found on the Internet.
- Learns a compact discriminative representation with few parameters on relatively small datasets.
- Allows for efficient comparisons by Euclidean distances.

We complement our in-depth quantitative analysis with visualizations for qualitative analysis. In addition to our feature extraction network learning approach we present a novel visualization approach for comparing image similarity between two images by exploiting the change in the extracted features when partially occluded.

2. Related Work

Fashion: Interest in fashion has been growing in the computer vision community. Some of the more traditional problems have been semantic segmentation of garments [26, 41, 43], image retrieval [12, 15], and classification of garments and styles [2, 5, 16, 35, 42]. Attempting to directly predict more esoteric measurements, such as popularity [38] or fashionability [27], have also been recently studied. As defining absolute metrics is rather complicated in such a subjective domain as fashion, exploiting relative attributes [17] and learning image similarity [15, 34] have also been proposed. Many of these approaches rely on datasets created by crawling the Internet and have large amounts of exploitable weak labels [27, 39]. Yamaguchi *et al.* [39] used these tags to provide priors for semantic segmentation. Our method is complementary to these approaches and can provide features for greater performance.

Deep Learning: There has been a flurry of new research focusing on exploiting deep learning in computer vision. Much of it focuses on improving classification results on large datasets such as ImageNet [7] or Places [49]. This

has led from initial models such as Alexnet [18] to more sophisticated ones such as the VGG models [29] with up to 19 layers; or the Googlenet models [31], that jointly use convolutions of different sizes in each layer. A more structured analysis of different networks was presented by Chatfield *et al*. [4], where they also analyzed using bottleneck layers to provide features of different sizes. All these approaches rely on fully supervised datasets. Xiao *et al*. [37] extended learning to partially noisy labels. However, they still require roughly 50% of the labels to be correct and need to learn another network to correct the noisy labels, while only noisy labels suffice for our approach.

Deep Similarity: Instead of learning classification networks, it is possible to directly learn similarity using deep neural networks. A popular approach consists of Siamese networks [3], in which a pair of inputs is used simultaneously to train a neural network model. The loss encourages similar inputs to have similar network outputs and dissimilar inputs to have dissimilar network outputs. This method has been recently applied with great success to local feature descriptors [10, 28, 45] and also for obtaining better representations of product images [1, 34]. It has also been extended to triplet images for ranking [11, 36] and has been very successfully applied to face recognition [25] in particular. We build upon this concept of image triplets for our ranking loss and show that by combining the ranking with classification results can be significantly improved.

Weak Data: We can identify two major sources of weak labels when using deep networks: text [9, 14] and image tags [22, 23]. Text can generally be found accompanying images and thus can be directly exploited as a form of weak label. Frome *et al*. [9] use text accompanying images to train more semantically meaningful classifiers, while Karpathy and Fei-Fei [14] use them as a form of weak annotation of the objects that lie in the image to perform localization. More recently, it has been seen that detectors seem to emerge when training deep networks for classification [48]. This has been utilized to learn models for semantic segmentation [22, 23]. However, as far as we know, we are the first to propose leveraging user-provided tags to learn discriminative features for a specific domain.

3. Method

We present a method for learning discriminative features from weakly-labeled data by jointly training both a feature extraction network and a classification network. A ranking loss on triplets of images is applied on the feature extraction network whose output is then fed into the classification network, where a classification loss is employed. After training, the feature extraction network can be used to provide discriminative features for other algorithms without a need for the classification network.

3.1. Joint Ranking and Classification

We formulate the problem as a joint ranking and classification problem. A ranking loss on triplets of images is applied on a feature extraction network, while a classification loss is employed on a classification network that uses the output of the feature extraction network. For training, we assume that we have a set of images with associated *weak labels* with large amounts of noise.

For ranking, we take three images as input simultaneously, as shown in Fig. 1. One image (center) is the anchor or reference image, to which the second image (right) is similar and the last image (left) is dissimilar. We assume that we have a similarity metric $r(\cdot,\cdot)$ between the weak labels of a pair of images. We consider two thresholds τ_s and τ_d that, given this metric, determine when two images are similar and dissimilar, respectively. Thus, two images I_1 and I_2, with labels $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ respectively, will be similar when $r(\boldsymbol{y}_1,\boldsymbol{y}_2) > \tau_s$ and dissimilar when $r(\boldsymbol{y}_1,\boldsymbol{y}_2) < \tau_d$. We will define each image triplet as $\mathcal{T} = (I_-, I, I_+)$ where $r(\boldsymbol{y}_-,\boldsymbol{y}) < \tau_d$ and $r(\boldsymbol{y},\boldsymbol{y}_+) > \tau_s$.

Suppose we have a set T of possible noisy *tags*, or attributes of an image like *red-sweater* or even just *red*. A *label* $\boldsymbol{l} = (l^t)_{t\in T}$ for an image assigns $l^t \in \{0,1\}$ to each tag t. If a tag applies to an image (*e.g*., if the tag is *red-sweater* and a red sweater appears in the image), the label for the image assigns 1 to the tag. Note that we assume these tags to be noisy and not exact. Let $|\boldsymbol{l}|$ be the number of tags that label $\boldsymbol{l}$ assigns 1. We propose using the similarity function between labels $\boldsymbol{a}$ and $\boldsymbol{b}$ defined as "intersection over union":

$$r(\boldsymbol{a},\boldsymbol{b}) = \frac{|\boldsymbol{a}\wedge\boldsymbol{b}|}{|\boldsymbol{a}\vee\boldsymbol{b}|}\,, \tag{1}$$

where $\wedge$ and $\vee$ operate on the labels as tag-wise minimum and maximum, respectively.

Given an image triplet $\mathcal{T}$, we use the feature extraction network to obtain a triplet of features $\mathcal{T}_f = (\boldsymbol{f}_-, \boldsymbol{f}, \boldsymbol{f}_+)$, where $\boldsymbol{f}$ is a feature vector of each image. We then compute a distance between two feature vectors and apply a ranking loss that encourages the distance d_+ between the anchor and the similar image to be smaller than the distance d_- between the anchor and the dissimilar image.

For comparing features, we consider the Euclidean distance $\|\cdot\|_2$. In contrast with [25], we normalize the distances instead of normalizing the feature pairs to have unitary norm. This changes the hard constraint into a soft constraint, which is essential for being able to learn using a classification loss and ranking loss simultaneously. We normalize the pair of distances (d_-, d_+) obtained from the feature triplet using the softmax operator:

$$d_- = \frac{\exp(\|\boldsymbol{f}_- - \boldsymbol{f}\|_2)}{\exp(\|\boldsymbol{f}_- - \boldsymbol{f}\|_2) + \exp(\|\boldsymbol{f}_+ - \boldsymbol{f}\|_2)} \tag{2}$$

$$d_+ = \frac{\exp(\|\boldsymbol{f}_+ - \boldsymbol{f}\|_2)}{\exp(\|\boldsymbol{f}_- - \boldsymbol{f}\|_2) + \exp(\|\boldsymbol{f}_+ - \boldsymbol{f}\|_2)}\,. \tag{3}$$

With distances now normalized to the $[0, 1]$ range, we define a ranking loss l_R that maximizes the dissimilar distance d_- and minimizes the similar distance d_+:

$$l_R(d_+, d_-) = 0.5\left((d_+)^2 + (1 - d_-)^2\right) = (d_+)^2 \ , \quad (4)$$

which is 0 only when $\|\boldsymbol{f}_+ - \boldsymbol{f}\|_2 = 0$ and $\|\boldsymbol{f}_- - \boldsymbol{f}\|_2 > 0$. In contrast to [36], the loss is normalized; because of that, we do not need to use large amounts of weight decay in order to ensure that the output of the network does not tend to infinity. In fact, we find that the implicit regularization provided by dropout and batch normalization are sufficient and do not rely on weight decay at all.

While the ranking loss l_R by itself should be sufficient to learn discriminative features, we found in practice that it is critical to complement it with a classification loss. We do this by employing a separate classification network that uses the features of the dissimilar image $\boldsymbol{f}_-$ and outputs a prediction value $X_- = (X_-^t)_{t \in T}, X_-^t = (X_{-,0}^t, X_{-,1}^t) \in \mathbb{R}^2$ for each binary value on each tag. We do not use the anchor image features $\boldsymbol{f}$ nor the similar image features $\boldsymbol{f}_+$, as they form a subset of the training images, unlike the dissimilar images, which are chosen randomly. With $\boldsymbol{y}_-$ as the noisy target label for the input image, we use multi-label cross-entropy loss for classification:

$$l_C(X_-, \boldsymbol{y}_-) = \frac{1}{|T|} \sum_{t \in T} l_\times(X_-^t, \ y_-^t) \ , \quad (5)$$

$$l_\times(x, y) = -x_y + \log\left(\exp(x_0) + \exp(x_1)\right) \ . \quad (6)$$

Finally, we combine both losses to obtain the model loss:

$$l(d_+, d_-, X_-, \boldsymbol{y}_-) = l_R(d_+, d_-) + \alpha l_C(X_-, \boldsymbol{y}_-) \ , \quad (7)$$

where α is a weight to balance the different loss functions.

The classification loss l_C affects both the feature extraction network and the classification network, while the ranking loss l_R only affects the feature extraction network.

3.2. Feature Extraction Network

We follow the approach of [29] of using 3x3 kernels for the convolutional filters to keep the number of weights down for the network and allow increasing the number of layers. One pixel padding is used to keep the input size and output size of the convolutional layers constant. In order to allow efficient learning the entire network from scratch, we rely on Batch Normalization layers [13]. Dropout [30] is used to prevent overfitting throughout the architecture.

A full overview of the architecture can be seen in Table 1. We note two important differences with commonly used networks: firstly, it uses a 3:4 aspect ratio for the input images as they are dominant in the fashion community; and secondly, it has a very small number of parameters compared to widely-used models. This is due to using only a single fully-connected layer with 128 hidden neurons and decreasing the number of filters before the fully-connected layer. This allows the model to have high performance while having only 1,572,544 parameters. In comparison, the VGG 16 layer network [29] has 134,260,544 parameters when considering only feature extraction.

Table 1: Feature extraction network architecture. All convolutional layers have 1×1 padding and all layers besides the max pooling layer have a 1×1 stride, while the max pooling layers have a 4×4 stride.

type	kernel size	output size	params
convolution	3×3	384x256x64	1,792
convolution	3×3	384x256x64	36,928
dropout (25%)		384x256x64	
max pooling	4×4	96x64x64	
batch normalization		96x64x64	128
convolution	3×3	96x64x128	73,856
convolution	3×3	96x64x128	147,584
dropout (25%)		96x64x128	
max pooling	4×4	24x16x128	
batch normalization		24x16x128	256
convolution	3×3	24x16x256	295,168
convolution	3×3	24x16x256	590,080
dropout (25%)		24x16x256	
max pooling	4×4	6x4x256	
batch normalization		6x4x256	512
convolution	3×3	6x4x128	32,896
fully-connected		128	393,344
TOTAL		128	1,572,544

3.3. Classification Network

Due to the noisy nature of the weak labels, the objective of the classification network is to aid the learning of the feature extraction network and not high classification performance. It consists of a batch normalization layer, followed by a rectified linear unit layer, a linear layer with 128 hidden units, and finally another linear layer which outputs the set of predictions X for classification. This network is kept small to encourage the propagation of gradients into the feature extraction network and hasten the learning. The initial batch normalization and rectified linear unit layers help partially isolate the classification network from the feature extraction network. When learning with 123 tags, the classification network has a total of only 48,502 parameters.

3.4. Joint Learning

Both networks are trained jointly using backpropagation [24]. Instead of using stochastic gradient descent which is dependent on setting a learning-rate hyperparameter, we utilize the ADADELTA algorithm [46], which adaptively sets the learning rate each iteration. No image cropping,

momentum, nor weight decay is used. The only image preprocessing consists of subtracting the mean from each color channel and dividing by the standard deviation.

Initialization is critical for learning both networks: even with batch normalization, we were unable to train both networks jointly from scratch. We overcome this issue by first training the feature extraction network with an additional fully-connected layer for classification (Eq. (5)). Once the optimization has converged, the additional classification layer is removed from the feature extraction network and the classification network is added with random weights. Finally, both networks are trained jointly.

Since it is impossible to precompute all the possible values of the similarity metric $r(\cdot, \cdot)$ for large datasets, we use a simple sampling approach for the triplet of images when using the ranking loss. We initially choose a random anchor image I. We then randomly sample an image I_r and check to see if $r(I, I_r) > \tau_s$ or $r(I, I_r) < \tau_d$. In the first case, I_r is added as the similar image I_+ and in the latter case it is added as the dissimilar image I_- to the image triplet. This is done until the image triplet is completely formed. If it is not formed in a set number of iterations, a new anchor image is chosen and the procedure is restarted.

4. Experimental Results

We implement our approach using the Torch7 framework [6]. We train our model on the Fashion144k dataset [27], and evaluate on the Hipster Wars dataset [16]. We compare our results against publicly available pretrained CNNs, and the state-of-the-art style descriptor [41] baselines. Our approach outperforms all baselines while being more efficient to compute and compact. We also perform additional experiments for the prediction of fashionability and see that we outperform all other approaches in the accuracy metric. In all cases, our joint classification and ranking approach outperforms using either classification or ranking losses alone, and using a Siamese architecture (described in the supplemental material).

4.1. Cleaning the Dataset

We train on the Fashion144k dataset [27]. Since the images have been obtained from chictopia.com without any sort of filtering, a large amount of images are not representative of what we wish to learn: a descriptor for fashion style. Thus, we would like to clean the data, *i.e.*, take only the images we consider suitable for training and not others. An example of images which we wish to classify are shown in Fig. 3. As it is unreasonable to do the cleaning manually, we train a classifier after a minor amount (6,000 images) of annotation that can be done in a couple of hours. We will show that this gives a significant increase in performance.

We annotate the images based on whether or not they contain a fully-visible person centered in the image (the supplemental material contains more details). We use a 1:1 train-to-test split to finetune the VGG 16 Layers Model [29] pretrained on ImageNet for the binary classification task of whether or not an image is suitable for training. We are able to obtain 94.23% accuracy on the 3,000 test images.

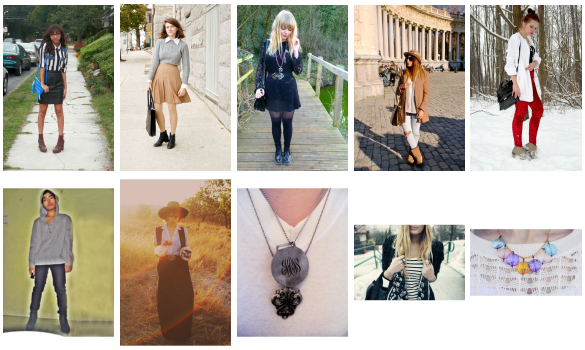

Figure 3: Examples of clean and dirty images from the Fashion144k dataset [27] are shown in the top and bottom rows, respectively. While there is much diversity, the clean images show figures more or less centered with the whole body visible, whereas dirty images have strong filters, show close-ups of objects, and/or are severly cropped.

As weak annotations, we use the "color" tags provided by the Fashion144k dataset which consist of *color-garment* tags such as *red-sweater* or *blue-boots*, the set of which we denote by T_1 and has 3,180 unique tags. We split the tags into colors and garments, resulting in a total of 123 unique weakly-annotated tags T_2. These tags are the only ones used when performing classification. However, for the weak metric (Eq. (1)), we consider the set formed by the union of the "color" tag set and the split tag set: $T = T_1 \cup T_2$.

We build a clean version of the Fashion144k dataset by first filtering out entries for which less than three tags in T_2 are assigned 1, to reduce the noise. We additionally filter images using our trained classifier on whether or not they are suitable. This results in 80,554 training images and 8,948 validation images with a 9:1 train to validation split.

4.2. Training the Model

We train with a batchsize of 32 for classification and 16 when jointly training for classification and ranking, due to the higher memory usage. We use a similarity threshold of $\tau_s = 0.75$ and dissimilarity threshold of $\tau_d = 0.01$. Examples of triplets of images $\mathcal{T}$ used for learning can be seen in Fig. 4. When jointly training for classification and ranking, we set the classification loss weight to $\alpha = 0.01$ such that the losses are of comparable magnitude. We initially train the feature extraction network with a classification loss. We then use the best performing model evaluated on the validation set to initialize the weights for the feature extraction network when using other losses. In particular, we consider joint classification and ranking loss, only ranking loss, and Siamese loss. We also compare to using the non-cleaned

Figure 4: Example of triplets of images used for training the model when using a ranking loss on the cleaned version of the Fashion144k dataset. For each triplet the anchor image I is displayed in the center with the dissimilar image I_- on the left and the similar image I_+ on the right.

dataset which we denote as "dirty".

4.3. Hipster Wars Dataset

We evaluate on the Hipster Wars dataset [16], which consists of similar images to the Fashion144k dataset [27] used to train, but from different sources. The dataset is made up of pictures of upright people in the center of the image; each corresponds to one of five styles: hipster, bohemian, goth, preppy, and pinup. The task is to perform 5-way clothing style classification. We compare against the state-of-the-art which is based on a 39,168-dimensional style descriptor [41] that is built by first estimating the 2D pose, which is trained in a supervised manner, and then extracting features for 32×32 pixel patches around all the pose keypoints. We also consider publicly available standard pre-trained networks on ImageNet and Places. All approaches except ours use fully-supervised noise-free data for training.

We evaluate the features by combining them with a linear SVM [8] with L_2 regularization and L_2 loss to predict the style. We perform 5-fold cross validation to set the regularization parameter and evaluate 100 times using random splits with a 9:1 train to test ratio as done in [16]. We consider the top $\delta = 0.5$ images from each style for classification. Each of the dimensions of the features are normalized independently using the training set such that the mean is 0 and the standard deviation is 1, except for approaches that directly learn embeddings, *e.g.*, Ranking, Joint, and Siamese models. We report accuracy, precision, recall, and intersection over union (iou) in Table 2. We can see that all our models outperform all the other approaches. By cleaning the data and improving the loss objective from classification to Siamese, Siamese to Ranking, and finally Ranking to Joint Classification and Ranking, we are able to improve performance.

We also consider two other scenarios: no training and fine-tuning. The results for not training and directly using feature distances is shown in Table 3. Our approach clearly outperforms other approaches, with a 20% increase in performance with respect to 4096-dimensional features and 50% increase with respect to similar size 128-dimensional features. If we use a single split instead of using 100 random splits and fine-tune the deep networks, we get the results shown in Table 4. Fine-tuning the deep network significantly improves the performance; however, the best performing network is still within 1% of our approach, despite using a $32\times$ larger internal feature representation.

Table 2: Comparison with the state-of-the-art on the Hipster Wars dataset. We evaluate as in [16] by computing the mean of 100 random splits with a 9:1 train to test ratio. For all the models we additionally display the number of parameters, and the dimension of the features. Dirty refers to training on a non-cleaned version of the Fashion 144k dataset. Our compact features significantly outperform the others.

feature	params	dim.	acc.	pre.	rec.	iou
Ours Joint	**1.6M**	**128**	**75.9**	**75.4**	**76.5**	**61.5**
Ours Ranking	**1.6M**	**128**	74.5	74.2	74.5	59.6
Ours Siamese	**1.6M**	**128**	73.3	72.9	74.0	58.2
Ours Classification	**1.6M**	**128**	73.5	71.7	74.1	57.3
Ours Joint Dirty	**1.6M**	**128**	72.9	72.1	73.1	57.0
Kiapour *et al.* [16]†	‡	39,168	70.6	70.6	70.4	54.6
VGG_CNN_M [4]	99M	4096	71.9	72.9	70.9	56.2
VGG 16 Layers [29]	134M	4096	70.1	70.5	69.7	54.8
VGG_CNN_M_1024 [4]	86M	1024	70.4	71.1	69.5	54.2
VGG_CNN_M_128 [4]	82M	**128**	63.5	62.8	63.5	46.3
VGG 16 Places [49]	134M	4096	57.4	57.6	59.4	41.5

† We were unable to reproduce the results of [16] and instead compare with the results from the confusion matrix they provide in their paper.
‡ Not directly comparable but in the order of hundreds of thousands.

Table 3: Comparison with deep networks using feature distances on the Hipsters Wars dataset. For each image in the dataset, we sort all the remaining images by distance and consider a top-n match if one of the n nearest images is of the same class. No training is done at all.

feature	dim.	top-1	top-2	top-3
Ours Joint	**128**	**63.5**	**79.9**	**86.3**
VGG_CNN_M [4]	4096	53.2	71.7	81.3
VGG 16 Layers [29]	4096	53.2	71.5	80.4
VGG_CNN_M_128 [4]	**128**	44.6	64.0	76.2
VGG 16 Places [49]	4096	40.1	61.0	72.0

4.4. Predicting Fashionability

We also evaluate on the much more complicated task of fashionability prediction on the Fashion144k dataset [27]. This consists of rating how fashionable a person in an image is on a scale of 1 to 10. As this is the dataset used for training, although with a different objective, we use only

Table 4: Comparison against fine-tuned deep networks on the Hipster Wars dataset. For the fine-tuned networks, the numerator is the fine-tuned result and the denominator is the result of using a logistic regression without fine-tuning.

feature	dim.	acc.	pre.	rec.	iou
Ours Joint	**128**	68.4	66.1	67.9	51.0
VGG_CNN_M	4096	$^{68.4}/_{64.6}$	$^{67.3}/_{64.2}$	$^{68.8}/_{63.0}$	$^{51.8}/_{46.8}$
VGG 16 Layers	4096	$^{63.8}/_{63.3}$	$^{62.6}/_{62.6}$	$^{63.5}/_{61.9}$	$^{46.5}/_{45.4}$
VGG_CNN_M_128	**128**	$^{62.6}/_{57.2}$	$^{60.4}/_{55.1}$	$^{62.1}/_{56.9}$	$^{44.5}/_{39.0}$

Table 5: Results on the Fashion144k dataset for the task of predicting how fashionable a person is on a scale of 1 to 10. We compare against strong CNN baselines and all variations of our model. We also consider our model architecture with random weights to observe the effect of learning, and average the results for 10 random initializations. The model with random weights performs almost the same as the VGG_CNN_M_128 model that has been pretrained on ImageNet. Our model outperforms all models in accuracy.

feature	dim.	acc.	pre.	rec.	iou
Ours Joint	**128**	**17.0**	14.7	15.2	7.1
Ours Classification	**128**	14.6	12.7	14.5	6.3
Ours Siamese	**128**	13.9	11.9	**24.2**	5.8
Ours Random	**128**	13.0	10.8	11.5	4.9
VGG 16 Layers [29]	4096	16.6	**15.1**	15.7	**8.0**
VGG 16 Places [49]	4096	15.8	14.0	14.7	7.3
VGG_CNN_M [4]	4096	13.2	11.8	11.5	6.0
VGG_CNN_M_128 [4]	**128**	13.2	10.8	11.7	4.8

the images not used in the training set for evaluation. We use 8,000 images for training and 948 images for testing. As with the Hipster Wars dataset, we evaluate the features using a linear SVM with L_2 regularization and L_2 loss and set the regularization parameter with 5-fold cross validation.

We compare against deep network baselines in Table 5. We can see our approach is able to significantly outperform the 128-dimensional feature network. However, it is outperformed by some of the 4096-dimensional features from deep networks. This is likely due to the fact that the fashionability score, while correlated with the style of the outfit, is greatly affected by non-visible factors such as social connections [38], and thus larger features are beneficial. Despite this, our approach outperforms similar networks.

4.5. Visualizing the Style Descriptor

We follow a similar approach to [47] to visualize how the input image is related to our descriptor. Instead of focusing on a single neuron, however, we consider the entire style descriptor output by displaying both the norm and projecting it onto PCA basis. We do this by sliding a 48×48 bounding box around the input image with the mean color of the input

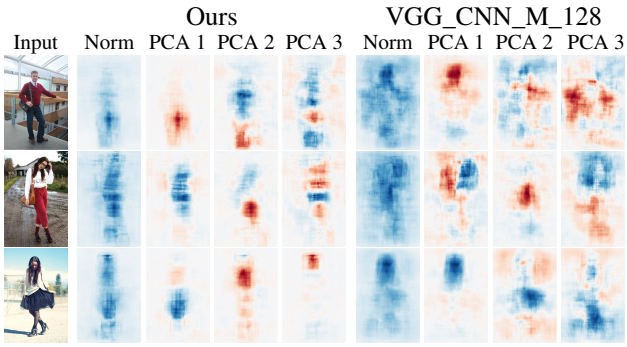

Figure 5: We analyze the relationship between the image and the style descriptor by moving an occluding box around the image and display the change in the norm of the descriptor and the change of the first three components on PCA basis computed on all the vectors extracted on the image. The positive and negative values are encoded in blue and red respectively. The norm is normalized so that the minimum value is white and the maximum value is blue, while the PCA representations are normalized by dividing by the maximum absolute value. Large descriptor changes correspond to the location of the individual and the PCA modes refer to the location of different garments. We compare with a fine-tuned VGG_CNN_M_128 network and see that our approach focuses on the figure and not the background.

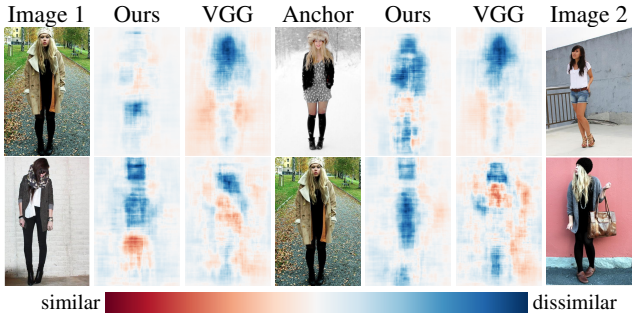

Figure 6: We analyze how the style of two image matches by using our descriptor. Blue indicates that the area is dissimilar between the images using the middle anchor image as a reference, while red represents similar areas in the image. The distance between the descriptors are shown above each visualization map. Our approach is capable of finding similarities between the fashion styles in the images. We compare with a fine-tuned VGG_CNN_M_128 network which can't identify the clothing changes in the image.

image and calculating the style descriptor. We compare the resulting descriptors with the original image descriptor and visualize the change. In this way, we can localize the parts of the image that have the greatest effect on the descriptor, *i.e.*, the parts that our feature extraction network focuses on. We show examples on the Fashionista dataset [40] in Fig. 5 and see that our style descriptor reacts strongly to different parts of the body for different individuals.

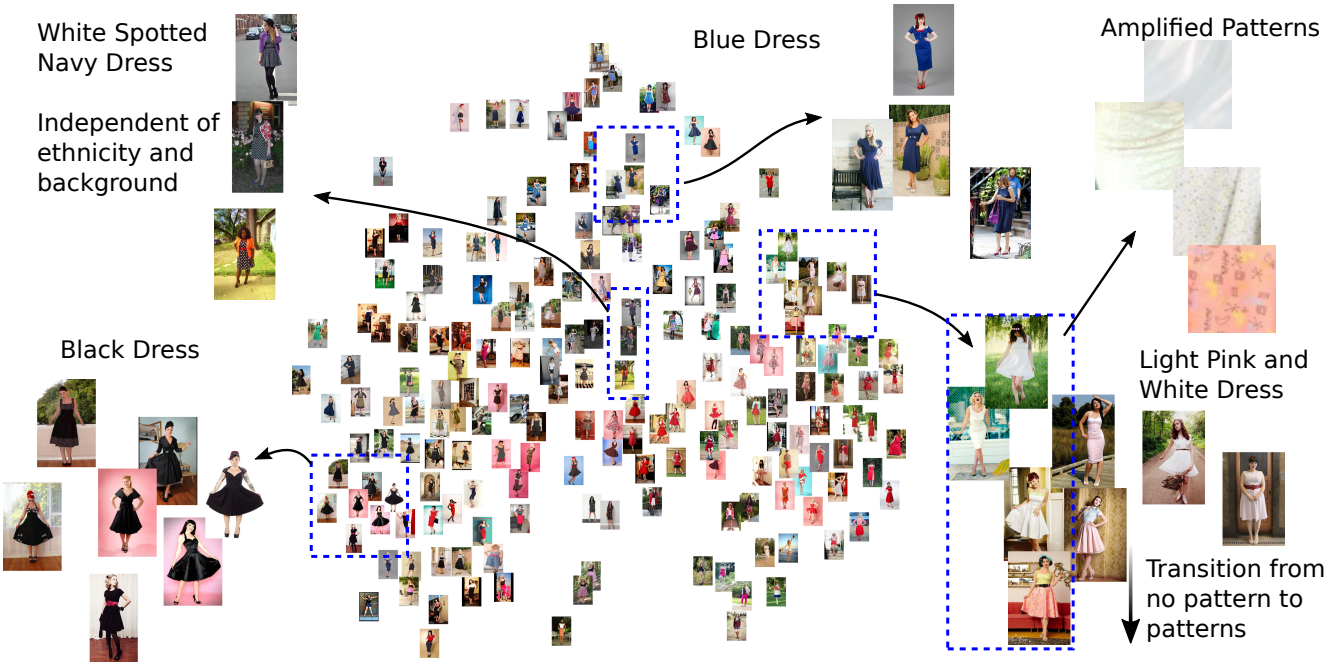

Figure 7: Visualization of the fashion style space of the Pinup class from the Hipster Wars [16] dataset using t-SNE [33].

4.6. Matching Styles

Instead of considering only a single image, we can consider pairs of images and match parts of the image using the style descriptor. This is done in a similar way as visualizing a single style descriptor, that is, by sliding a 48×48 pixel bounding box around the image. The difference is that we consider two feature descriptors $\boldsymbol{f}_1 = f(I_1)$ and $\boldsymbol{f}_2 = f(I_2)$ corresponding to two different images I_1 and I_2 simultaneously. We employ the difference between both feature vectors to evaluate how well this vector matches the change of the style descriptor $f(\cdot)$ given an image partially occluded at pixel location (u, v) with a bounding box mask $B(u, v)$ by using the dot product:

$$I_M(u, v) = (f(I_1 \circ B(u, v)) - \boldsymbol{f}_1) \cdot (\boldsymbol{f}_2 - \boldsymbol{f}_1) \ , \quad (8)$$

where $I_M(u, v)$ is the output map at pixel (u, v), and $\circ$ is the Hadamard product or element-wise matrix multiplication.

We show results in Fig. 6 where we can see that our descriptor is effectively capturing the notion of outfit similarity in a reasonable way. Note that this concept of local outfit similarity was learned automatically from noisy user-provided tags without any pixel level annotations. On the other hand, the fine-tuned VGG_CNN_M_128 model gives similar maps regardless of the image compared to: it is overfitting to the Hipster Wars dataset.

4.7. Exploring the Fashion Style Space

Finally, we perform a qualitative analysis of the resulting style descriptors obtained by visualizing the fashion style space using t-SNE [33]. The style descriptors can be compared efficiently by using Euclidean distances. We visualize the Hipster Wars "Pinup" class in Fig. 7. Our features display a remarkable robustness to background changes and focus on the outfit. They are also able to capture subtleties such as the transition from pink dresses without patterns to floral patterns, and group navy dresses with white spots regardless of the background and the wearer's ethnicity.

5. Conclusions

We have presented a novel approach to weakly-supervised learning of features consisting of joint ranking and classification. This allows learning compact 128-dimensional features for more specific types of images that may be very costly and complicated to annotate. Our method allows us to learn discriminative features that are able to outperform both the previous state of the art and the best-performing model trained on ImageNet while being the size of a SIFT descriptor. The proposed joint ranking and classification approach consistently improves results over using either classification or ranking loss alone. We complement our model with a simple approach to automatically clean the data. In addition to a quantitative analysis, we present a new approach both to visualize the individual descriptor activations and to find similarities between two style images. Our analysis of the resulting descriptor shows it is robust to backgrounds and is able to capture fine-grained details such as flower patterns on pink dresses.

Acknowledgements: This work was partially supported by JSPS KAKENHI #26108003 as well as JST CREST.

References

[1] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Trans. on Graphics (SIGGRAPH)*, 34(4), 2015. 3

[2] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. V. Gool. Apparel classifcation with style. In *ACCV*, 2012. 2

[3] J. Bromley, I. Guyon, Y. Lecun, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. In *NIPS*, 1994. 3

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 3, 6, 7

[5] H. Chen, A. Gallagher, and B. Girod. Describing clothing by semantic attributes. In *ECCV*, 2012. 2

[6] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 5

[7] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008. 6

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 3

[10] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015. 3

[11] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *ICLR*, 2015. 3

[12] J. Huang, R. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV*, 2015. 2

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[14] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 3

[15] M. Kiapour, X. Han, S. Lazebnik, A. Berg, and T. Berg. Where to buy it: Matching street clothing photos in online shops. In *ICCV*, 2015. 2

[16] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. Hipster wars: Discovering elements of fashion styles. In *ECCV*, 2014. 2, 5, 6, 8

[17] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Interactive image search with relative attribute feedback. *IJCV*, 115(2):185–210, 2015. 2

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3

[19] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015. 1

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 2

[21] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 1

[22] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? – Weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015. 3

[23] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015. 3

[24] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. In *Nature*, 1986. 4

[25] F. Schroff, D. Kalenichenko, and J. Phibin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 3

[26] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. A High Performance CRF Model for Clothes Parsing. In *ACCV*, 2014. 2

[27] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. Neuroaesthetics in Fashion: Modeling the Perception of Fashionability. In *CVPR*, 2015. 2, 5, 6

[28] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative Learning of Deep Convolutional Feature Point Descriptors. In *ICCV*, 2015. 3

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 4, 5, 6, 7

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15:1929–1958, 2014. 4

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 3

[32] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 1

[33] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9:2579–2605, Nov 2008. 8

[34] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *ICCV*, 2015. 2, 3

[35] S. Vittayakorn, K. Yamaguchi, A. C. Berg, and T. L. Berg. Runway to realway: Visual analysis of fashion. In *WACV*, 2015. 2

[36] 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*, 2014. 3, 4

[37] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015. 3

[38] K. Yamaguchi, T. L. Berg, and L. E. Ortiz. Chic or social: Visual popularity analysis in online fashion networks. In *ACMMM*, pages 773–776, 2014. 2, 7

[39] K. Yamaguchi, M. H. Kiapour, and T. L. Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *ICCV*, 2013. 2

[40] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012. 1, 2, 7

[41] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Retrieving similar styles to parse clothing. *PAMI*, 2014. 2, 5, 6

[42] K. Yamaguchi, T. Okatani, K. Sudo, K. Murasaki, and Y. Taniguchi. Mix and match: Joint model for clothing and attribute recognition. In *BMVC*, 2015. 2

[43] W. Yang, P. Luo, and L. Lin. Clothing co-parsing by joint image segmentation and labeling. In *CVPR*, 2014. 2

[44] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014. 1

[45] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015. 3

[46] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012. 4

[47] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 7

[48] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015. 3

[49] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 1, 2, 6, 7