Efficient Hierarchical Embedding for Learning Coherent Visual Styles

Ruining He, Xiaodan Wu, Shengye Wang
UC San Diego
9500 Gilman Drive M.C. 0404
La Jolla, California
{r4he, xiaodanwu, shengye}@cs.ucsd.edu

1. INTRODUCTION

The visionary Steve Jobs said, "A lot of times, people don't know what they want until you show it to them." A powerful recommender system not only shows people similar items, but also helps them discover what they might like, and items that complement what they already purchased. In this paper, we attempt to instill a sense of "intention" and "style" into our recommender system, i.e., we aim to recommend items that are visually complementary with those already consumed. By identifying items that are visually coherent with a query item/image, our method facilitates exploration of the *long tail* items, whose existence users may be even unaware of.

This task is formulated only recently by Julian *et al.* [1], with the input being millions of item pairs that are frequently viewed/bought together, entailing noisy style coherence. In the same work, the authors proposed a Mahalanobis-based transform to discriminate a given pair to be sharing a same style or not. Despite its success, we experimentally found that it's only able to recommend items on the margin of different clusters, which leads to limited coverage of the items to be recommended. Another limitation is it totally ignores the existence of taxonomy information that is ubiquitous in many datasets like *Amazon* the authors experimented with. In this report, we propose two novel methods that make use of the hierarchical category metadata to overcome the limitations identified above. The main contributions are listed as following.

- We focus more on learning coherent visual styles that go cross *different* subcategories; that is we aim to learn (for instance) what kind of shoes go with a pair of pants. Therefore unlike [1], we only care about item pairs that connect different subcategories.
- We propose two efficient methods to smoothly incorporate category information, which are (1) suitable to extract various style dimensions from different subcategories, and (2) helpful to mitigate the limited coverage problem suffered by existing work.
- Experiments on large real-world dataset demonstrate that our methods are able to achieve state-of-the-art prediction accuracy, while significantly reduce the training time by more than $5\times$. Due to tight time constraints and the large size of the dataset we experimented with, we were unable to fine-tune the hyperparameters of our models, but we do believe even larger improvements can be achieved if given enough time.

In the e-commerce world we are currently living in, almost all major companies are using recommender systems to recommend items that are of potential interests to their customers. The saving in computation and improvement in prediction accuracy translate directly to millions of dollars in sales.

2. RELATED WORKS

Applying image search and image similarity comparison has been explored both in academia and industry. Google shopping and Like.com are some well-known applications of image-based fashion recommendation applications. However, the limitation of such applications is that they fail to capture human notion of "je ne sai quoi", a consistent style shared among objects from very different categories.

Other visual based recommendation system utilizes a large amount of text, metadata and human-curated data. Linaza, Garcia(2013) [3] described an image-based travel recommendation system that allows travelers to specify their interests through a set of images, from which the system infers their profile. Jing, Liu and Kislyuk (2015) [4] demonstrated that content recommendation powered by visual search improves user engagement. Bell and Bala (2015) [5] trained a convolutional neural network to identify products in scene and find stylistically similar products. Simo-Serra et al. [6] predict the fashionability of a person in a photograph and suggest subtle improvements. Jagadeesh et al. [7] use a street fashion dataset with detailed annotations to identify accessories whose style is consistent with a picture. Another method was proposed by Kalantidis, Kennedy and Li [8], which accepts a query image and uses segmentation to detect clothing classes before retrieving visually similar products from each of the detected classes. Among these, McAuley et al. [1] is closest to what we propose here; it uses visual features extracted from convolutional neural networks and learn a visual similarity metric to identify substitute and complementary items to a query image. What's worth reminding ourselves is that any recommender system that relies on user explicit input can be attacked or vandalized, such as shilling attacks and deliberate mistagging. Our approach has the advantage of doing away with annotations.

We seek to measure visual distance among objects from different categories, thus helping to fulfill recommender system's essential purpose, to discover things we didn't know before. The novelty factor is a very important aspect of the recommendation problem. It has been acknowledged that providing obvious recommendations can decrease user satisfaction. We address the problem of finding the right trade-

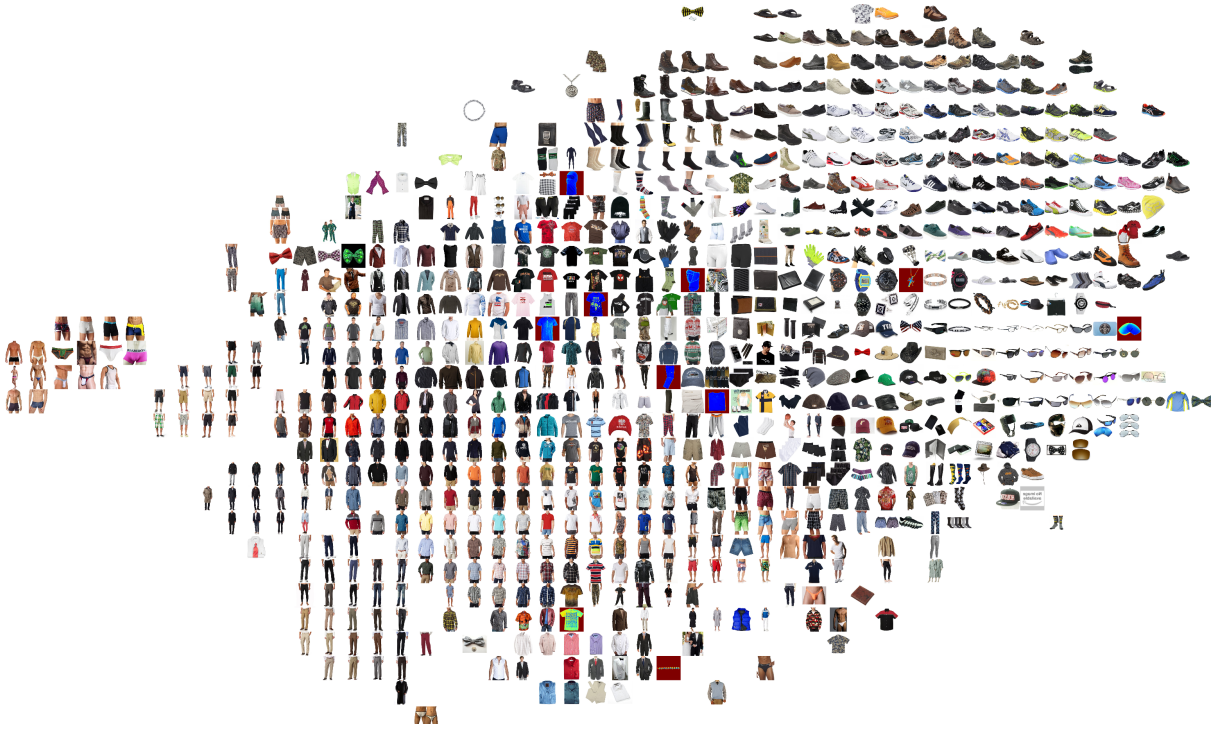

Figure 1: A t-SNE [2] visualization of the embedded 'style space' learned on *Amazon Men's Clothing* by the model proposed by McAuley *et al.* [1]. One limitation of this method is it tends to learn *clusters* of items. If we were to recommend outfits that are consistent with a given query image, there is an unwanted tendency to recommend those items that are on the margins of different clusters.

off between finding novelty items and keeping high quality recommendations.

3. BACKGROUND

The field of deep learning using Convolutional Neural Networks (CNNs) has made amazing progress over the last decade in recognizing objects across wide baselines and wide changes in appearance. CNNs consist of layers of small computational units that process visual information hierarchically in a feed-forward manner. Each layer of units extracts a certain level of features from the input image.

McAuley et al. [1] proposed to make use of a pre-trained CNN to extract a F-dimensional feature vector for each item in the dataset. Afterwards, they used a low-rank approximation of the Mahalanobis transform to embed the image space to a 'style space' where items with similar styles are projected to nearby locations. This approach yields good accuracy in predicting related items. However, we experimentally observed two major shortcomings:

1. Figure 1 illustrates the uncovered 223-D visual space by [1][1], further embedded to 2-D by t-SNE [2] for visualization purpose. Note that this method projects different categories to be clusters in the 'style space'. This leads to a few unexpected results: (1) We want to recommend outfits, which essentially requires us to explore different categories for a given query image. However, [1] will only able to recommend items on the *margin* of different clusters/categories. (2) Items to the far end or lie in the middle of each cluster are mistakenly assumed to be "not consistent with all other types of items".

2. [1] is using only one embedding matrix, which is assumed to be *universally* applicable to every item across all subcategories. However, this assumption is questionable. For example, the definition of causality of pants and sweater is different: a casual pair of pants may have many pockets while a casual sweater may have a hood. Therefore, using a single embedding matrix is limited by its expressive power by being only able to uncover characteristics that are shared by all subcategories.

[1]Trained in our Experiment Section as baseline.

4. OUR APPROACHES

To compensate the above shortcomings of using a single embedding matrix, we propose two novel methods that take category information into consideration: Category-aware Embedding (CAE) and Sparse Hierarchical Embedding (SHE). CAE utilize category information by assign an embedding matrix for each of the categories, and SHE reduces the number of parameter to train by sharing them on a hierarchy basis. We compare CAE and SHE with the baseline [1].

4.1 Category-aware Embedding

This first method is Category-aware Embedding (CAE). CAE is relatively straightforward based on the observation that category information should affect embedding matrix, therefore it assigns a unique embedding matrix for each leaf

(i.e., the finest subcategory) on the category tree. It is expected that CAE will improve prediction accuracy, however it will also limit the number of embedding dimensions considering training efficiency and the number of parameters we can afford.

Embedding matrix (or transform matrix) $\mathbf{U}$ projects items into the style space such that items with similar style are mapped to nearby locations. More specifically, each row of $\mathbf{U}$ extracts one characteristic (attached to a dimension of the style space) from the F-dimensional CNN feature f_i of item i. The baseline [1] projects i from the feature space to the following point s_i in the style space:

$$s_i = \mathbf{U} f_i. \tag{1}$$

Considering our task aiming to learn coherent styles from different categories, a single embedding matrix is limited to extract the same characteristic from different subcategories. For example, imagine that we are extracting a characteristic (dimension) that corresponds to the notion of 'casual' across two subcategories: shoes and shirts. Since shoes that are colorful are more casual than those with a single color, we would require a vector in $\mathbf{U}$.

Such intuition leads us to the idea of representing different categories with different transform matrices. To implement such idea, we choose a certain layer of subcategories on the category tree, assign each with a transform matrix U_k. That is, we use the following equation to map from CNN feature space to the style space:

$$s_i = \mathbf{U_k} f_i, \quad \text{where item } i \text{ belongs to category } k \tag{2}$$

We call such scheme Category-aware Embedding (CAE).

4.2 Sparse Hierarchical Embedding

CAE improves accuracy by introducing the idea of assigning a transform matrix for each of the subcategories, at the cost of more parameters and therefore slows down the training process. Because for n leaf subcategories, there are $n \cdot F$ parameters to learn, where F is number of dimension of CNN feature space. To mitigate this problem, we propose another model — Sparse Hierarchical Embedding (SHE) which exploits the hierarchical structure of the category tree.

Figure 2 illustrates the basic idea of SHE. In this instance, we allocate the dimensions in the hierarchical tree using a 4-3-2-1 distribution of rows in the embedding matrix. For example, for a "Walking Shoe" subcategory, its transform matrix consists four rows from "Men" (i.e., the root node), three rows from "Shoes" (i.e., grandfather node), two rows from "athletic" (i.e., father node) and one row from itself. Note that in this way each subcategory is attached with a 'collective' embedding matrix (with 4+3+2+1 = 10 dimensions). Different subcategories are sharing much 'embedding vectors' from their common ancestor nodes.

Briefly speaking, SHE aims to share the embedding vectors across the embedding matrices of different subcategories (leaves). This enables us to afford more embedding dimensions and fully make used of the hierarchical structure. By sharing the embedding vectors that are attached to the common ancestors, it significantly reduces the number of required parameters and largely improves training efficiency.

Importantly, it is worth pointing out that SHE is extremely *expressive*. When the dimension allocation is $K : 0 : \cdots$, SHE can reduce to baseline with K embedding dimensions. And when the allocation is $0 : 0 : \cdots : K : 0$, SHE can reduces to CAE with K dimensions as well. This flexibility gives us more opportunity to exploit the hierarchical structure and can be extremely helpful to deal with issues like sparsity, unbalanced training pair distribution, and so forth.

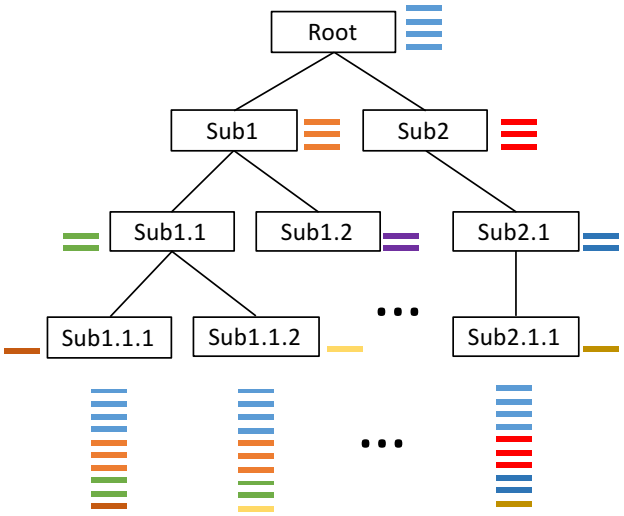

Figure 2: Illustration of the 4-3-2-1 allocation (i.e., ten embedding dimensions) of SHE on a 4-layer category tree. Each bar represents a row in the 'collective' embedding matrix associated with the leaf node. Items from different subcategories are efficiently sharing the same embedding vectors attached to their common ancestors.

4.3 Training the Models

As with the training procedure in [1], we use logistic regression to model the log-likelihood of our training corpus, consisted of both positive and negative pairs. Due to limited space and the similarity between our training procedure and that of the baseline, interested readers can read [1] for more details. We use $\mathcal{L}_2$-norm regularization of all parameters to avoid over-fitting. Regularization hyperparameters are tuned with grid search to be described later in the Experiment section.

5. EXPERIMENTS

We implement all models with C++ and run all our experiments on a desktop machine with Intel Core i7-4900MQ processor and 24 GB main memory. In this section, first we introduce the dataset we experimented with by demonstrating basic statistics, then we describe the experiment setting and baselines before we show and analyze our experimental results.

5.1 Dataset Statistics

Due to time constraints, we focus on a subcategory — Men Clothing — of the *Amazon Clothing & Accessories* dataset introduced by Julian *et al.* [1]. In order to make use of the hierarchical structure of this subtree, we further go down three layers, which gives us in total 163 categories (i.e., leaves in the subtree) and 785,419 training pairs after we dropped those pairs connecting items from a same sub-

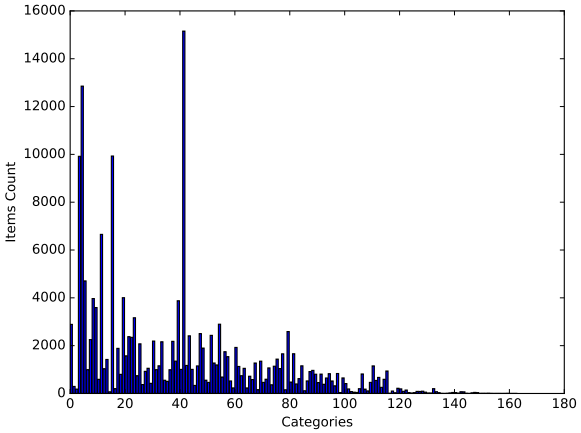

Figure 3: Number of items in each subcategory.

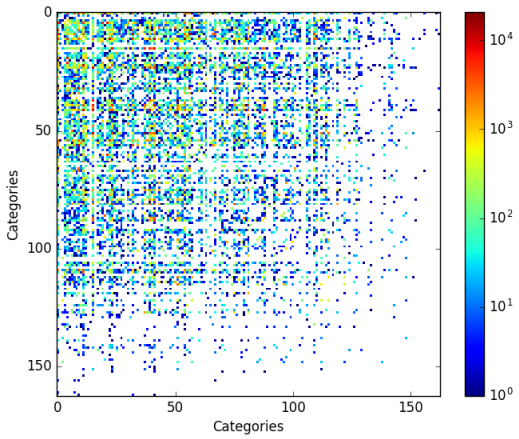

Figure 4: Number of training pairs connecting each subcategory pair, demonstrated with a heat map.

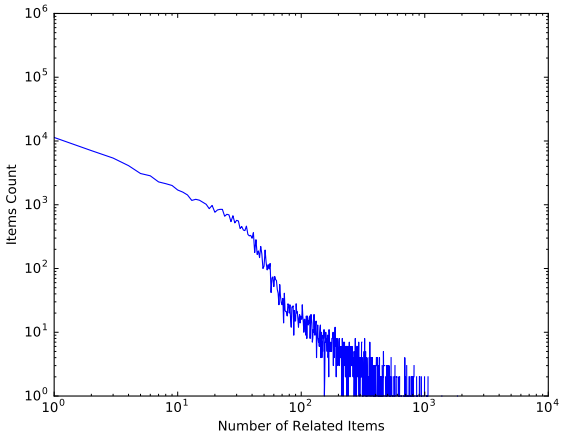

Figure 5: Degree distribution of all items in our Men Clothing dataset (log-log plot).

category. Note that this subset we use is still large enough to be representative as there are still more than 1.5 millions of pairs (positive + negative) for our experiment. Following [1], we use the CNN features of each item extracted from a pre-trained CNN as f_i.

Figure 3 shows the number of items in each of the 163 categories we have, each of which is indexed by a category ID. As we can see from this figure, the popularity of categories are not uniformly distributed. With this information in mind, we can't afford two many parameters for each subcategory due to the existence of sparse subcategories.

All our item pairs are connecting different subcategories. It's helpful to demonstrate how such pairs are distributed over different subcategory pairs. Figure 4 uses a heat map to show such distribution. From the figure it can be seen that some subcategory pairs are more heavily connected than others. This drives us to come up with a baseline that simply use this statistics. In fact, the authors of [1] already included such a baseline in their paper and demonstrated that it can't outperform the Mahalanobis transform-based method which we use as the main baseline in this report.

Due to the popularity differences between different items, some items may have more links than others. Figure 5 demonstrates the degree distribution of different items with a log-log plot. As we can see from this figure, it approximately conforms to the Power-law distribution.

5.2 Implementation Details

In our experiment, we use the 'also-bought' metadata provided by Amazon as our training data. It entails noisy style compatibility of item pairs and are statistically large enough to be appropriate for learning the notion of coherent styles. As introduced earlier, we only keep those pairs that are connecting different subcategories on the deepest layer of the category tree. In our case, we limited the height of the category tree to be four (including the root node).

5.2.1 *Sample negative pairs*

It requires a set of negative pairs with equal size to the positive pairs to train all models. We sample the negative set strategically, maintaining its degree distribution to be the same to that of the positive set.

5.2.2 *Comparison Methods*

We mainly compare with the most related work with ours, which is also current state-of-the-art method to perform the same task.

- **Baseline:** Proposed by McAuley *et al.* [1], this method uses a single embedding matrix to learn the style space. It's state-of-the-art model for learning visually coherent styles.

- **CAE:** This is the first method proposed by this report, which learns a unique embedding matrix for each subcategory. Since there are too many leaves on the fourth layer, we finally associated a embedding matrix to each subcategory on the third layer.

- **SHE:** This is the second method proposed by this report, which fully exploits the hierarchical structure of the category tree. It's so flexible as to be able to associate different number of dimensions to different layers.

	Baseline (223-d)	CAE (8-d)	SHE (10-d)	CAE Improvement	SHE Improvement
Error Rate	7.732%	7.4062%	8.4261%	4.40%	-8.23%
Time Consumption	61.57 hrs	10.58 hrs	4.97 hrs	581.94%	1238.83%

Table 1: Comparison between different methods under the same total number of parameters. Baseline benefits from using a much higher dimensional style space (223-d), while CAE and SHE only use around 10 dimensions.

	Baseline (10-d)	SHE (10-d)	SHE Improvement
Error Rate	13.06%	8.42%	55.10%

Table 2: Comparison between baseline and SHE under the same number of embedding dimensions (10-d).

5.2.3 *Experiment Setting*

To make fair comparisons between different models, we mainly compare under the same total number of parameters. For SHE, we allocate the embedding dimension distribution over different levels by 4-3-2-1 in the hierarchical tree, as shown by Figure 2. This yields 10 embedding dimensions in total. This also introduces 223 embedding vectors in total, which sets the standard for our comparison.

For CAE, we have 27 subcategories in total on the third layer, therefore we use 8 embedding dimensions for each subcategory, which gives us approximate 223 embedding vectors (i.e., $27 \times 8 = 216$ vectors).

We get our training/validation/test subsets by using a 80/10/10 split of the full dataset. Regularization hyperparameters for all models are tuned with grid search to perform the best on the validation set, and in all cases we report the corresponding performance on the test set. All our experiments use the same training procedure with [1], i.e., we use the library L-BFGS to learn all our parameters simultaneously.

5.3 Experimental Results

Table 1 demonstrates experimental results we got for the experiment setting described earlier. From the table we can see that our methods are able to achieve competitive or better accuracy at a much lower training cost.

More precisely, CAE significantly improved the error rate by 4.40%, with even less number of parameters ($8 \times 27 = 216$ vs. 223 embedding vectors). Meanwhile, it consumes much less time: the time reduction is 5.81×. This demonstrates that:

1. Taking category information into account can help improve the accuracy significantly. Therefore, it is important and beneficial to incorporate such data in recommender system.

2. Using category information also makes the training more efficient, as we only need a small fraction of embedding dimensions (10 vs. 223) to achieve comparatively good results. This also reduces training cost significantly as we only need to update much less parameters for each training pair (i.e., those associated with the subcategories the two items fall into).

On the other hand, SHE reduces the training time consumption by 12.39×, although it yields slightly worse error rate. There are a few possible explanations for its accuracy degradation:

1. SHE is using a much lower dimensional style space than the baseline (223 v.s. 10), which may limit the expressive power of SHE. To validate this hypothesis, we performed another group of comparison between SHE and baseline, as shown in Table 2. From this table we can see that under 10 embedding dimensions, SHE actually beats baseline by as much as 55.10%. Therefore, when both of the algorithms use the same dimensions in the embedded style space, SHE in fact yields much better results.

2. The allocation of dimensions on the category tree is not fine-tuned. Current allocation 4-3-2-1 is only randomly assigned by hand. There is great room to fine-tune this allocation to improve the accuracy.

3. We may have over-fitted the embedding vectors on the deepest level as there are many sparse subcategories which only contains very few items and thereby are associated with few training pairs. Unfortunately, we don't have enough time to validate this hypothesis due to tight time constrains considering the size of datasets we experiment with.

5.4 Visualization

It's helpful to visualize our learned style space and see if our proposed methods can project items with similar styles to nearby locations. To this end, we use t-SNE [2] to embed the learned 8-d style space into 2-d, preserving the relative distance information between different items. Figure 6 demonstrates our style space learned by CAE on our Men Clothing dataset. As we can see from this figure, our method works successfully to learn coherent styles across different subcategories as it can project different visually compatible items to nearby places within the style space.

Compared to the style space revealed by the baseline [1] (i.e., Figure 1), our model can better 'blend' different categories to a certain degree which can mitigate the 'limited coverage' challenge confronted by existing methods as we identified earlier.

6. CONCLUSION

In this report, we presented two novel models to efficiently learn the notion of visual compatibility across different subcategories by exploiting the hierarchical category information. Context-aware Embedding utilizes category information by assigning an embedding matrix to each of the subcategories, while SHE further reduces the number of parameters to learn by fully exploit the hierarchical structure to facilitate sharing parameters. Experimentally, we found that category information is very useful for the learning task. We achieved competitive or even better prediction accuracy at

Figure 6: Illustration of a t-SNE [2] 2D visualization of the embedded style space uncovered by our model. Items are a random sample comprised of 7000 images from the test set. Compared to the style space revealed by the baseline [1], our embedding can 'blend' different categories to a certain degree which can mitigate the challenge confronted by the baseline as we identified earlier.

a much lower training cost than the state-of-the-art method. In spite of our promising results, there is still much room to further improve our approach be by fine-tuning our hyper-parameters (i.e., the dimension allocation). With a dataset as huge as ours, we don't have sufficient time to push our proposed methods to their full strengths.

7. ACKNOWLEDGMENTS

We would like to take the opportunity to thank Prof. Julian, Sheeraz Ahmad, Daryl Lim, etc. for their dedicated work in CSE 255, Fall 2015.

8. REFERENCES

[1] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *SIGIR*, 2015.

[2] L. van der Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of machine learning research*, 2014.

[3] A. Garcia, I. Torre, and M. T. Linaza, "Mobile social travel recommender system," in *Information and communication technologies in tourism 2014*, pp. 3–16, Springer, 2013.

[4] Y. Jing, D. Liu, D. Kislyuk, A. Zhai, J. Xu, J. Donahue, and S. Tavel, "Visual search at pinterest," *arXiv preprint arXiv:1505.07647*, 2015.

[5] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 98, 2015.

[6] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun, "Neuroaesthetics in fashion: Modeling the perception of fashionability," in *CVPR*, 2014.

[7] V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan, "Large scale visual recommendations from street fashion images," in *SIGKDD*, 2014.

[8] Y. Kalantidis, L. Kennedy, and L.-J. Li, "Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos," in *ICMR*, 2013.