Mining Fashion Outfit Composition Using An End-to-End Deep Learning Approach on Set Data

Yuncheng Li, LiangLiang Cao, Jiang Zhu, Jiebo Luo, *Fellow, IEEE*

***Abstract*—Fashion composition involves deep understanding of fashion standards while incorporating creativity for choosing multiple fashion items (e.g., Jewelry, Bag, Pants, Dress). In fashion websites, popular or high-quality fashion compositions are usually designed by fashion experts and followed by large audiences. In this paper, we aim to employ a machine learning strategy to compose fashion compositions by learning directly from the fashion websites. We propose an end-to-end system to learn a fashion item embedding that helps disentangle the factors contributing to fashion popularity, such as instance aesthetics and set compatibility. Our learning system consists of 1) deep convolutional network embedding of fashion images, 2) title embedding, and 3) category embedding. To leverage the multimodal information, we develop a multiple-layer perceptron module with different pooling strategies to predict the set popularity. For our experiments, we have collected a large-scale fashion set from the fashion website Polyvore. Although fashion composition is a rather challenging task, the performance of our system is quite encouraging: we have achieved an AUC of 85% for the fashion set popularity prediction task on the Polyvore fashion set.**

I. Introduction

Fashion style tells a lot about the subject's interests and personality. With the influence of fashion magazines and fashion industries going online, clothing fashions are attracting more and more attention. According to a recent study[1], the sales of woman's apparel in United States is $111 Billion in 2012 and keeps growing, representing a huge market for garment companies, designers, and e-commerce entities.

Different from well-studied fields including object recognition [1], fashion sense is a much more subtle and sophisticated subject, which requires domain expertise in outfit composition. Here an "outfit" refers to a set of clothes worn together, typically for certain desired styles. To find a good outfit composition, we need not only follow the appropriate dressing codes but also be creative in balancing the contrast in colors and styles. Fig. 1 is an example showing the nontrivial nature of fashion outfit composition. The outfit composition includes a Felicia Lace mix dress, a Nylon backpack with print, in addition to a pair of Nicholas Kirkwood Scarp Glitter Mary Janes shoes. Normally people do not pair a fancy dress with a casual backpack, however, once the shoes were is in the outfit, it completes the look of a nice and trendy outfit.

Fig. 1: An example showing the challenging of fashion outfit composition. Normally one would not pair a fancy dress (as in the left) with the casual backpack (as in the bottom right) but once you add in the shoes (as in the top right), it completes the look of a nice outfit.

Although there have been a number of research studies [2] [3] [4] on clothes retrieval and recommendation, none of them considers the problem of fashion outfit composition. This is partially due to the difficulties of modeling outfit composition: On one hand, a fashion concept is often subtle and subjective, and it is nontrivial to get consensus from ordinary labelers if they are not fashion experts. On the other hand, there may be a large number of attributes for describing fashion, for which it is very difficult to obtain exhaustive labels for training. As a result, most of the existing studies are limited to the simple scenario of retrieving similar clothes, or choosing individual clothes for a given event.

This paper proposes a data-driven approach to train a model that can automatically predict suitable fashion outfit compositions. This approach is motivated by the recent surge of online fashion communities, including Polyvore, Pinterest, and YouTube videos, which have greatly helped spreading fashion trends and fashion tips, creating an online culture of sharing one's style with other Internet and mobile users. Such online communities can be very big. For example, Polyvore received 20 million unique monthly visitors in May 2014. By exploiting the online fashion community, this paper collects a

This work was partly done when Yuncheng Li was interned at Yahoo Research

Y. Li is with the Department of Computer Science, University of Rochester, Rochester, USA e-mail: yli@cs.rochester.edu

L. Cao is with the Yahoo Research, NYC, USA e-mail: liangliang@yahoo-inc.com

J. Zhu is with the Yahoo Inc., Sunnyvale, USA e-mail: jiangzhu@yahoo-inc.com

J. Luo is with the Department of Computer Science, University of Rochester, Rochester, USA e-mail: jiebo.luo@gmail.com

Manuscript received XXXXX; revised XXXX.

[1]http://goo.gl/QdrAvQ

large dataset of popular fashion outfits , named “Fashion Sets”, which are image collages curated and shared by community members. Our dataset contains hundreds of thousands of clothes with metadata information, which is associated with user engagement for every set. The popularity of a fashion set depends on the aggregated opinion of the crowd, which averages out the complicated context factors and correlates better with the fashion set itself. We believe a machine can learn the intelligence of this difficult task by leveraging the wisdom from millions of users.

Our approach aims to solve two main challenges:

1) Complicated visual contents of a fashion image. There are potentially many attributes corresponding to different visual aspects: categories, colors, coherence, patterns, general pairing codes, creative pairing choices, as well as styles for different ages and individuals. It is impossible to label or even list all possible attributes for every clothing image.
2) Rich contexts of fashion outfit. Clothing outfits can reflect personality and interests. Clothing may reflect the style and tradition of different brands and designers, or even the culture of a specific group. To infer such information, we must take into account not only the pixel information but also the context information in the fashion outfit.

To solve the first challenge, we propose an end-to-end system of encoding visual features using a deep convolutional network, which can take a fashion set as the input and predict whether it will be a popular set. To address the second challenge, we propose a multi-modal deep learning framework, which exploits the context information from image, title and category. As the experiments show, the multi-modal approach significantly outperforms a single modality, and provides a more reliable solution for fashion set composition.

In summary, our contributions are three folds:

- We present a novel problem: fashion set popularity classification. Predicting fashion set popularity is an extremely challenging problem, because many interleaving factors, visible or hidden, contribute to the process. The combinatorial nature of the problem also makes it very interesting to serve as a test stone for the state of the art machine learning systems. We also show that by learning to predict the fashion set popularity, the system can be used to build novel outfit compositions, by fashion item retrieval and importance ranking based on set compatibility.
- We propose an end-to-end trainable system to fuse signals from multi-level hybrid modalities, including the images and metadata of the fashion items, and the information across the fashion items.
- We collect a large-scale dataset for the fashion set related research. This dataset contains fashion items and sets, associated rich context information, and will be released for future research by the community.

II. Related work

Fashion domain is a very important and lucrative application of computer vision[5], [6]. The majority of research in this domain focus on fashion image retrieval[7], [8], [9], and fashion image attribute learning[4], [10], [11], [12]. There are also studies on evaluating the compatibility between fashion items[3], [2]. Specifically, Veit et al. proposed to learn clothing matches from the Amazon co-purchase dataset [3], and Iwata et al. proposed a topic model to recommend “Tops” for “Bottoms”[2]. Comparing with the previous works on fashion image retrieval or clothing style modeling, the goal of this work is to predict fashion set popularity, which has its own challenges in modeling many aspects of the fashion item sets, such as compatibility and aesthetics.

The techniques developed in this paper belong to the category of image set classification. Unlike our work which focuses on fashion set, most of the exisiting work was applications of face recognition from video frame sequences and multi-view object recognition. Recently, Huang et al. proposed a parametric manifold distance based on tangent map, and used metric learning to learn a problem adaptive manifold metric to perform kNN classification [13]. Lu et al. proposed a novel loss function to train Convolutional Neural Networks for the image set classification problem[14]. Li et al. combined dictionary learning and metric learning to recommend online image boards[15]. The key differences of this work from previous image set classification are the following: 1) We are the first to consider fashion image sets. The goal of the fashion image set modeling is to disentangle the style factor hidden among the fashion images, while the previous works[13], [14] focuses on the cases where the images share the same object class. 2) Along with the images, we integrate the metadata of the fashion items to further improve the fashion set modeling, for which we propose to jointly learn modality embedding and fuse modalities.

Our work is partially motivated by the previous work in multimedia understanding [16], [17], [18], [19], [20], [21], [22]. Most of these works suggest to leverage visual analysis with other modalities such as text and audio information. Moreover, the recent progress of deep neural networks in visual recognition [23], [24], [25] and natural language processing [26], [27], [28] has shown that the recognition performance in both fields have been greatly improved. Inspired by these progress, this paper tries to validate that we can achieve better results by combining the-state-of-the-art techniques in both fields.

III. Dataset

We collect a dataset from Polyvore.com, the most popular fashion oriented website based in US. Polyvore provides tools, templates and friendly interfaces for users to create fashion sets, and ttens of thousands of fashion sets are created everyday. Fig. 1 shows examples of user curated fashion sets. As shown in the example, the fashion items are carefully and beautifully organized to demonstrate specific fashion styles. These fashion sets are viewed, favored and recreated by visitors, and some of the fashion sets attract high volume of attention. Fig. 3a and 3b are fashion set examples. From Polyvore.com, we crawl fashion sets, which are accociated with the number of likes and a number of fashion items. For each fashion item, we crawl the image, the title, and

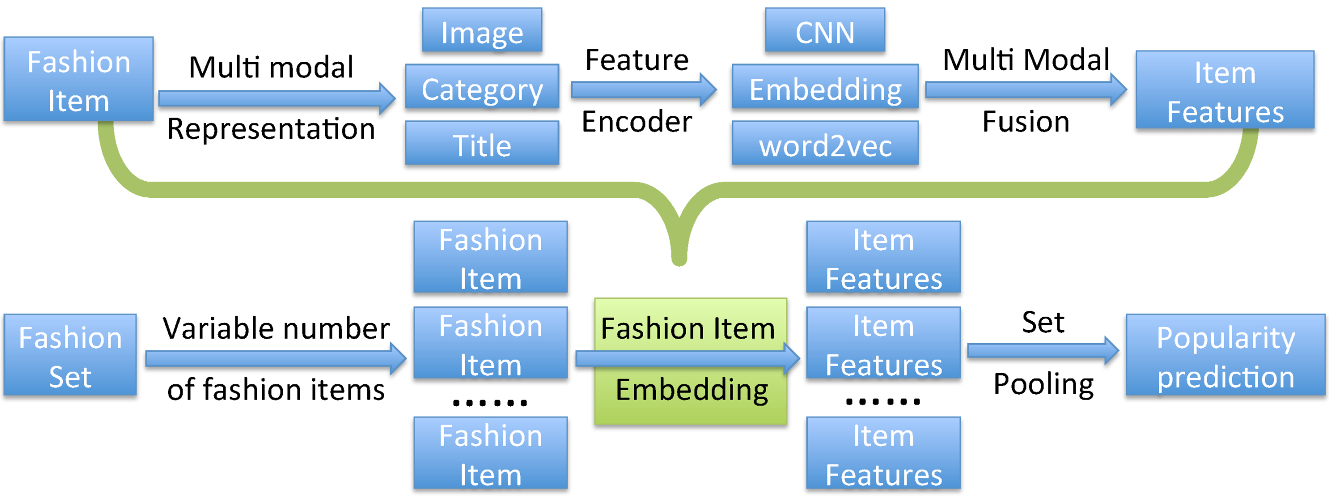


Fig. 2: The proposed framework for fashion set popularity prediction.

(a) A popular set

(b) An unpopular set

the category. As an example, the dress shown in Fig. 1 is accociated with the title "Self-Portrait Felicia Lace Mix Dress", the category "Cocktail Dresses" and the image.

We perform some simple filtering over the crawled datasets to clean up the raw dataset. (1) Each fashion item is associated with a category, such as "Flip Flops" and "Men's Bracelets". We remove the categories associated with fewer than 500 items. (2) Remove items that appear in more than 5 sets. This is to avoid the model overfitting to a particular item. (3) Given the like count for the fashion sets, we obtain the 1^{th}, 40^{th}, 90^{th} and 99^{th} percentiles. We label the sets with A like count between 1^{th} and 40^{th} as unpopular, and those with A like count between 90^{th} and 99^{th} as popular sets. Therefore, we have four times more unpopular sets than popular sets. We throw away sets that do not fall into either range, because they are either outliers or uncertain sets. (4) We segment the entire fashion sets into training, development, and testing splits. For this segmentation, we make sure there is no overlap between any two splits, so that the items in the testing split are never seen in the training. To achieve this, we construct a graph with the fashion sets as nodes, and if two fashion sets have a common item, we draw an edge between the corresponding nodes. After graph segmentation based on connected components, we obtain the fair training/development/testing splits. (5) To simplify data pipeline, we fix the number of items in the fashion set as 4. For sets with more than 4 items, we random sample up to 5 random combinations and treat them as independent 4 item fashion sets.

	train	dev	test	total
#sets	156,384	19,407	195,71	195,262
#items	294,782	366,77	367,92	368,251

TABLE I: Basic statistics for the training/development/testing split.

After such pre-processing, the number of fashion sets and the fashion items are listed in Table I for each training/development/testing splits. After the filtering process, the total number of item categories is 331. The most frequent category has 18,407 items, while the least frequent category has only 21 items. The category examples and the number of items statistics are shown in Fig. 4.

IV. Methodology

A. Problem Formulation

Let S_i denote a set, and x_{ij} denote the items in the set S_i, so that $x_{ij} \in S_i$. Let x_g denote an arbitrary item in the database. Each item x_g is associated with multiple modalities, such as image, title and category. Let x_g^k denote the k^{th} modality of the item x_g. We learn a model $f(S_i)$ to predict whether the set S_i will become popular. The prediction model consists of (1) feature encoders for individual modalities $E^k : x_g^k \mapsto R^d$, where d is the feature dimension; (2) a fusion model to integrate the multiple modalities $F : x_g \triangleq \{E^k(x_g^k), k =$

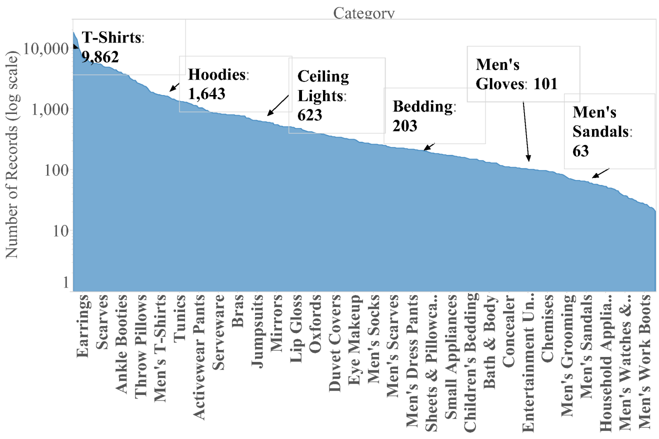


Fig. 4: Category frequency and category examples. The Y-axis is in the logarithmic scale.

Notation	Meaning
S_i	a fashion set
x_{ij}	a fashion item in the set S_i
x_g^k	the k^{th} modality of the fashion item x_g
$f(S_i)$	predicted popularity score for set S_i
$E^k(x_g^k)$	feature encoder for the k^{th} modality
$F(x_g)$	feature fusion model for the fashion item x_g
$P(S_i)$	feature pooling model for the fusion set S_i
$C(S_i)$	popularity polarity classification model for set S_i

TABLE II: Definition of the notations

$1 \dots K\} \mapsto R^d$, where K is the total number of modalities (in this paper, $K = 3$); (3) a pooling model to map the set into a single feature vector $P : S_i \triangleq \{x_{ij}, j = 1 \dots |S_i|\} \mapsto R^d$; (4) a classification model to perform prediction $C : P(S_i) \mapsto \hat{Y}_i \in \{0, 1\}$;

Table II list all the notations for easy reference. Fig. 2 gives an overview of the fashion set popularity prediction pipeline. These components are connected and trainable from end to end, and the training set is a list of pairs (S_i, Y_i), where Y_i is the label telling if S_i is popular ($Y_i = 1$) or not ($Y_i = 0$). In the following sections, we explain each component separately.

B. Visual Feature Encoding

As shown in the fashion item example (Fig. 1), the fashion item image contains the most important information needed for outfit composition, and we encode the image with Convolutional Neural Networks (*ConvNets*).

ConvNets are used to encode the item image x_g^1. There are many ConvNets architectures to choose from, and we use a variant of AlexNet for simplicity. The AlexNet variant is also composed of 5 convolution layers with relu and 1 linear layer with relu and dropout, but we use batch normalization in place of the local response normalization layer, because batch normalization shows more stable results in various tasks. We use the output of the *fc6* layer as the encoding feature of the input image. The dimension of the image encoding is 4096. Because of the changes made into the AlexNet, and because of the very different domain, we do not use any pretrained weights.

C. Context Feature Encoding

While images contains the most important information for outfit composition, the state of the art image understanding fails to capture all the details in the images for composition inference. We employ the context information, including the item title and category as a remedy to help fashion item modeling.

GloVe model is used to encode the item title x_g^2 [27]. *GloVe* model is a distributed word representation that maps words to a 300d vector space. The vector space captures the word semantics, and has been used in a variety of text analysis applications. Given an item title, (1) a basic tokenizer is used to chop it into tokens; (2) the tokens are mapped to 300d vectors using the *GloVe* model; (3) average the vectors into a single 300d vector. For simplicity, we employ simple average pooling to aggregate the GloVe vectors, while there are more sophisticated methods using Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN) to pool GloVe vectors [29].

Categorical embedding is used to encode the item category $x_g^2 \in \{1 \dots CC\}$, where CC is the total number of categories. Basically, *Categorical embedding* is a lookup table that maps the item category (represented by integers) into a trainable vector. The dimension of the category embedding is 256.

D. Leveraging multiple information

Multiple modalities contain complementary information for the final task. In the following, we explain our pipeline to fuse the multiple modalities.

Given the encoded feature vectors $\{E^k(x_g^k)\}$ for a fashion item x_g, (1) using a single layer perceptron, reduce the dimension of the feature vector $E^k(x_g^k) \in R^{d_k}$ to the same size d: $E^k(x_g^k) \mapsto \tilde{E}^k(x_g^k) \in R^d$; (2) concatenate the vectors $\{\tilde{E}^k\}$ into a single vector $R^{K \times d}$, where K is the total number of modalities ($K = 3$ in this paper); (3) apply 2-layer MLP (multiple layer perceptron) with RELU nonlinearity and dropout to map the concatenated vector to R^d.

Using the feature encoders and the fusion model, the fashion items are embedded into the d dimensional vector space. A fashion set is composed of multiple fashion items, i.e., $S_i \triangleq \{x_{ij}, j = 1 \dots |S_i|\}$, and the pooling model P maps the set into a single vector. In practice, the pooling function should handle variable length S_i, and respect the fact that the input is an order-less set. In the experiments, we find that non-parametric element wise reduction works very well, while satisfying the requirements and being very efficient. The non-parametric reduction $r : \{p \in R\} \mapsto R$ is based on commutative functions, such as *add*, *max* or *prod*. For example, the mean reduction r^{mean} takes a set of numbers and outputs the *mean* value at each dimension, the maximal reduction r^{max} takes a set of numbers and output the *maximal* value. Applying the reduction function r for each dimension independently, the vector set $\{R^d\}$ is mapped to a single vector R^d.

The pooling model P aggregates information from individual fashion items to produce a holistic embedding for the fashion set. Empirically, we observed that the element wise reduction function works very well. However, there are other more advanced models that work very well to deal with variable length inputs, such as Recurrent Neural Networks (RNN)[30]. In the following, we explain the RNN based pooling models.

Recurrent Neural Networks (RNNs) achieved the state of the art performance on image/video captioning, language modeling, and machine translation, by modeling the dependency within time series. We adapt RNN as a pooling model to encode the variable length fashion items. Specifically, given a fashion set $S_i \triangleq \{x_{ij}, j = 1 \cdots |S_i|\}$, RNN performs the following operations iteratively.

$$h_i^t = tanh(h_i^{t-1} W_h + x_{it} W_x + B), \tag{1}$$

where h^0 is set to all zero vector, x_{it} is the embedding of the fashion item (given by the feature encoders and fusion models), W_h, W_x and B are model parameters, and $h_i^{|S_i|}$ is used as the embedding of the fashion set S_i.

E. End-to-end learning

Given the feature encoders, fusion model and pooling model, the fashion set S_i is mapped to a fixed length vector $s_i \in R^d$, which is used in the following linear model to make popularity prediction,

$$Pr(\hat{y}_i = 1 | S_i) = \delta(w^T s_i + b), \tag{2}$$

where the w^T and b are the weights and bias of the linear model, respectively, and $\delta(s)$ is the sigmoid function $1/(1 + exp(-s))$.

To learn the parameters in the framework, a loss function l_i, based on cross entropy, is defined on the training data (S_i, Y_i),

$$l_i = Y_i Pr(\hat{y}_i) + (1 - Y_i)(1 - Pr(\hat{y}_i)) \tag{3}$$

As shown in the Fig. 2, all these models are connected together and the gradients can be backpropagated to learn the model parameters, so we end up with a trainable system to learn all the components adaptively. The learnable parameters in the system includes the convolution and linear layer weights in the image encoder $E^1(x_g)$, the categorical embedding vectors in the item category encoder $E^3(x_g)$, the weights of the MLP layers in the fusion model F, the parameters in the pooling model P and the linear classification model C. In particular, in the item title encoder, we use GloVe model pretrained on webpage corpus with 42 billion tokens and 1.9 million unique words, and we fix it during the training.

V. Experiments

Given the dataset with training/development/testing splits, we learn the model parameters using the training split, selecting hyperparameters using the development split, and report the performance on the testing split. We use the area under curve (AUC) of the ROC curve and the average precision (AP) as the evaluation metric to select hyperparameters and compare performance. In this section, we perform a comparative analysis to validate the proposed fashion set popularity prediction model. There are a few hyperparameters in the proposed models, such as the modality combination, the embedding dimension d, the pooling model P and the number of iterations. In the following sections, unless specified otherwise, we analyze each of them separately, and when analyzing a specific parameter, we use the development split to select the other parameters and report the performance on the testing split. For the optimization parameters, we use the Adam optimizer with a batch size of 50, learning rate of 0.01, and 40K iterations. We reduce the learning rate in half for every 15K iterations.

A. Comparison with existing methods

There are two directions weakly related to this work: image set classification based on metric learning [13], and clothing style learning using Siamese network[3]. In the following, we explain how we adapt these methods for the fashion popularity prediction task, and make comparison with our proposed method.

The LEML method, proposed in [13], is a recent algorithm on image set classification. The traditional image set classification methods assume the images in a set reside in a structured manifold, and proposed various manifold metrics based on the image feature co-variance. Specifically, LEML is a metric learning method to learn distances between image set pairs S_i and S_j. Given tuples of (S_i, S_j, y_{ij}), where $y_{ij} \in \{0, 1\}$ tells whether S_i and S_j are from the same class ($y_{ij} = 1$ if yes), LEML learns a distance metric $d(S_i, S_j)$, such that $d(S_i, S_j)$ is small for $y_{ij} = 1$, and $d(S_i, S_j)$ is large for $y_{ij} = 0$. We refer to the original paper [13] for more details, such as the parametric forms of the $d(S_i, S_j)$ and the optimization objectives. The complexity of the original LEML algorithm is in the order of the number of training samples, so we need to adapt it using stochastic approximation to make it work with the large scale problem presented in this paper. In addition, LEML algorithm works on static features, so we need to extract visual features from the images. For this purpose, we extract the widely used semantic features using the AlexNet pretrained on the ImageNet object recognition task[1]. Given a testing fashion set, the nearest neighbor training samples are retrieved in the learned metric space, and the labels of the neighbors are voted to obtain the popularity label for the testing fashion set.

There are existing systems on using end-to-end trainable ConvNets to match clothing images[3] using the Siamese architecture. Siamese architecture can also be used in the popularity prediction task. Specifically, given a training sample (S_i, Y_i), we construct the tuple (Q_i, t_i, Y_i), where the t_i is an arbitrary item from the set S_i, and $Q_i \subset S_i$ is the query set, such that $t_i \notin Q_i \wedge \{t_i\} \cup Q_i = S_i$. Each tuple (Q_i, t_i, Y_i) constitute a query and target pair. By assuming that the fashion set popularity stems from the item matching within the set, the popularity label Y_i also tells whether Q_i and t_i is a good match. With these notations, the following contrastive loss is minimized to train the ConvNets,

$$\begin{aligned} l(Q_i, t_i, Y_i; \theta) &= Y_i d^2 + (1 - Y_i) max(\alpha - d, 0)^2 \\ d = d(Q_i, t_i; \theta) &= \|Q_i(\theta)) - t_i(\theta)\|, \end{aligned} \tag{4}$$

	LEML[13]	Siamese[3]	Ours (image only)
AUC	0.616	0.601	**0.757**
AP	0.237	0.169	**0.364**

TABLE III: Comparison with existing methods.

where θ are the weights in the ConvNets and pooling models, and α is the margin to control sensitivity (fixed at 10 in the experiments). Given a testing fashion set S_i, we pick an arbitrary item as the target t_i, and the remaining items as the query Q_i, while the negative distance $-d(Q_i, t_i; \theta$ is used as the popularity confidence. For fair comparison, we use the same number of iterations and optimization method for all ConvNets based methods, including the *Siamese* baseline, and the propose framework.

In Table III, we compare the proposed models with the LEML model and Siamese architecture. To make a fair comparison, we use only images in this experiment. From this result, we make the following observations: (1) Compared with LEML, which is a recently published image set classification method based on manifold metric learning, our proposed framework benefits from end-to-end representation learning. Once a good representation is learned, it becomes straightforward to aggregate the evidence from the fashion sets. (2) Compared with the Siamese network, our proposed framework solves the popularity prediction problem directly by classification, and thus outperforms the Siamese network, which solves the popularity prediction problem indirectly by matching query and candidates.

B. Different modalities

	AUC	AP
image	0.757	0.364
category	0.778	0.343
image+category	0.818	0.414
title	0.820	0.455
image+title	0.822	0.476
title+category	0.835	0.476
image+title+category	**0.852**	**0.488**

TABLE IV: Comparison of different modality combinations.

The item image, title and category are available for the fashion set popularity prediction. In this section, we compare the strength of each modality and their combinations to predict fashion set popularity. Given a total of three modalities, there are seven modality configurations. The performance is compared in Table IV.

As shown in Table IV, fashion item title works best on its own, but image and category add complementary information. As shown in Fig. 4, many items are labeled as very high level category, such as "clothing", which partially explains why category alone performs poorly. Fig. 5 shows the convergence property with different modality combinations. The models converge well, except for the *Siamese* baseline, which is commonly known to converge slowly[3].

C. Comparison of different pooling model P

The performance is compared in Table V, where *mean* denotes the element wise mean reduction on the feature

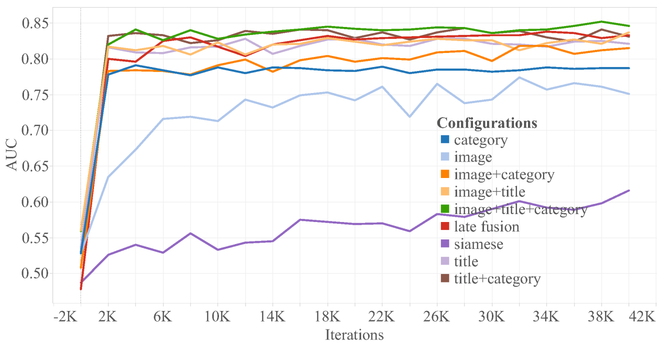


Fig. 5: Convergence property with difference modalities. (Best viewed in color)

	mean	max	rnn
AUC	**0.852**	0.839	0.813
AP	**0.488**	0.476	0.416

TABLE V: Comparison of different pooling models.

vectors, *max* denotes the maximal reduction, and *rnn* denotes the RNN based pooling model.

As shown in Table V, mean reduction achieved the best result. RNN performs poorly, because it enforces a specific order over the fashion items, but the fashion set is essentially orderless.

D. Comparison with late fusion

In this section, we investigate the importance to consider fashion sets collectively at the training time. Specifically, we train single image based popularity classifier by propagating the set labels to the member items. During the testing time, the popularity scores of individual items are aggregated using mean or max to obtain the set level popularity score. The late fusion model ignores the dependency among the items during the training time. As shown in Table VI, the late fusion model performs poorly compared with our method, which illustrates the importance to consider the item dependency and to model the items collectively.

E. Comparison over the embedding dimension d

Using the feature encoders and fusion model, the fashion items are embedded into a fixed length vector of dimension d. The performance with different embedding dimensions are listed in Table VII, which shows that the performance peaks at $d = 8$. Too small or large embedding size will hurt the performance.

	max	min	mean	ours
AUC	0.797	0.812	0.836	**0.852**
AP	0.374	0.406	0.444	**0.488**

TABLE VI: Comparison with late fusion models

	1	2	4	8	16	32	64	128
AUC	0.813	0.834	0.840	**0.852**	0.841	0.841	0.842	0.848
AP	0.43	0.452	0.452	**0.488**	0.487	0.484	0.465	0.467

TABLE VII: Performance with different embedding dimensions d

F. Retrieval experiments

One motivation to study fashion set popularity is to help curators build popular fashion sets. It is common to build the fashion sets in an item by item manner, and the problem can be formulated as the following algorithm:

- Initialize the empty set $S_0 = \{\}$.
- Retrieve the top ranked item t_i:

$$t_i = \arg\max_{t_i} Pr(\hat{y}_i = 1 | \{t_i\} \cup S_i) \tag{5}$$

- Increase set size: $S_{i+1} = \{t_i\} \cup S_i$.
- Repeat until S_i is satisfied.

The core of this algorithm is the ranking problem in (5). In this section, we analyze the ranking capability of different models. Specifically, given a popular fashion set S_i, we pick one item arbitrarily as the target r_i, and the remaining items $Q_i \triangleq S_i - \{t_i\}$ as the query fashion set. For each target t_i, we random sample four items as additional candidates N_i. Therefore, we formulate the following ranking problem for each popular fashion set,

$$t_i = \arg\max_{t_i \in N_i \cup \{r_i\}} Pr(\hat{y}_i = 1 | \{t_i\} \cup Q_i) \tag{6}$$

In order to compare the ranking capabilities of different models, we resort to two different labeling processes, automatic labeling and manual labeling. Both labeling processes provide the positive candidates P_i for each query Q_i, and the positive candidates are used to compute the *recall@1*:

$$\textit{recall@1} = \frac{1}{T} \sum_i t_i \in P_i \tag{7}$$

For each popular fashion set S_i, the automatic labeling process: $P_i^0 = \{r_i\}$, labels only the actual target as positive. For manual labeling, we resort to fashion experts to label each of the combinations $Q_i \cup t_i|_{t_i \in \{r_i\} \cup N_i}$, so that the positive candidate set P_i can contain more than the actual target r_i. The automatic labeling process gives very large sample size easily, but the manual labeling is more accurate because a random sample can be positive as well. For the evaluation purpose, our fashion experts labeled 150 queries, and for each query Q_i, there are 5 candidates $N_i \cup \{r_i\}$, and one of the 5 candidates is the actual target r_i. The fashion experts labeled 93.7% of the 150 actual targets as compatible with the corresponding query, which validates the use of set popularity as the fashion item compatibility measure. On the average, two items are labeled by fashion experts as positive for each query set.

While asking the fashion experts to label the matches, we also ask them to explain the decisions. There are certain factors frequently mentioned by the experts, for example, color tone, color contrast, pattern prints, categorical matches, and coherence. Ideally, all these factors can be captured by the image alone, but due to the limitation of the current image understanding capabilities, the metadata is more powerful for recognizing certain fashion factors, such as the categorical matches and pattern prints.

Table VIII shows the ranking examples and *recall@1* using different modality configurations. From the examples and *recall@1* evaluation scores, we make the following observations:

- While the *full* configuration performs the best with the popularity prediction task, the image alone configuration performs the best in the ranking experiments, partially because there is a discrepancy between the loss function used in the popularity prediction task and the ranking task.
- Overall, the performance of the ranking experiments needs to improve further. In particular, using automatic labeling, the random guess baseline has a *recall@1* of 0.2, but none of the methods is better than it. While using manual labeling, the performance is slightly better than random guesses, which has an expected *recall@1* of 0.4.

G. Fashion item importance

A fashion set consists of multiple fashion items, and these items contribute differently to the fashion set popularity. In this section, we analyze the item importance in predicting the fashion popularity. This analysis can help fashion set curators to build more popular fashion sets, and also provides insights on what are actually learned by the models. Specifically, for a popular fashion set, (1) compute the popularity score of the fashion set; (2) replace an item in the set by a random item in the database, and thus build a new fashion set; (3) compute the popularity score of the new fashion set; (4) compute the popularity score decrement caused by the replacement; (5) order the items by the popularity score decrement.

The popularity decrease tells how important the fashion item contributes to the fashion set popularity. Table X shows the importance orders predicted by different modality configurations. These examples show that different modalities pick up different cues to predict popularities. It is useful to users by suggesting to replace the least significant items. For example, for the second set of Table X, the third item can be replaced because of its lowest importance score predicted by the *full* and *title* model, which makes sense because its color is not compatible with other items.

H. Error case analysis

In this section, we analyze the errors made by the models. We order the testing fashion sets by the popularity score predicted by the models. We investigate the top 1000 fashion sets that are actually not popular, i.e., the errors in the top 1000 predictions. The error examples and the error rates of various modality configurations are shown in Table IX, which shows that the trend of the error rates are comparable with Table IV, e.g., the full modality configuration produces the least errors in the top 1000 list. The errors made by the *image* model tend to be more visually consistent, and the errors made by the *title* and *category* tend to be more consistent in terms of the item semantics.

VI. Conclusions

In this paper, we consider the challenging problem of fashion set popularity prediction, which reflects the difficultities of matching domain expert knowledge and modeling the diversity in fashion. We propose an end-to-end trainable system for this

query	candidate	automatic label	manual label	methods
		5 3 4 2 **1** 2 3 4 5 **1** 2 5 3 4 **1** 2 3 4 5 **1**	**5** 3 **4** **2** 1 **2** 3 **4** **5** 1 **2** **5** 3 **4** 1 **2** 3 **4** **5** 1	image(0.13, 0.55) title(0.05, 0.40) category(0.10, 0.44) full(0.07, 0.43)
		1 4 2 5 **3** 1 2 4 5 **3** 1 2 5 4 **3** 1 4 2 5 **3**	**1** 4 2 **5** **3** **1** 2 4 **5** **3** **1** 2 **5** 4 **3** **1** 4 2 **5** **3**	image(0.13, 0.55) title(0.05, 0.40) category(0.10, 0.44) full(0.07, 0.43)
		5 1 3 4 **2** 1 5 4 3 **2** 1 3 4 5 **2** 1 5 **2** 4 3	5 1 **3** **4** **2** 1 5 **4** **3** **2** 1 **3** **4** 5 **2** 1 5 **2** **4** **3**	image(0.13, 0.55) title(0.05, 0.40) category(0.10, 0.44) full(0.07, 0.43)
		1 **3** 2 4 5 **3** 5 4 1 2 4 2 1 **3** 5 4 **3** 2 5 1	1 **3** **2** 4 5 **3** 5 4 1 **2** 4 **2** 1 **3** 5 4 **3** **2** 5 1	image(0.13, 0.55) title(0.05, 0.40) category(0.10, 0.44) full(0.07, 0.43)
		4 5 3 **2** 1 5 4 1 3 **2** 4 5 3 1 **2** 4 5 3 1 **2**	**4** 5 **3** 2 1 5 **4** 1 **3** 2 **4** 5 **3** 1 2 **4** 5 **3** 1 2	image(0.13, 0.55) title(0.05, 0.40) category(0.10, 0.44) full(0.07, 0.43)

TABLE VIII: Ranking examples and evaluation results. The first column are the query sets Q_i, the second column are the candidates $\{r_i\} \cup N_i$. The numbers following the candidates are the ranking score orders generated by the modality configuration, labeled by the "methods" column. In the "methods" column, "full" means combining all modalities. The positive candidates are highlighted under the labeling process names ("automatic label" and "manual label"). The numbers in the parentheses are the *recall@1* for the corresponding method with automatic labeling and manual labeling, respectively.

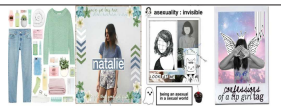
image	title	category	full
			
			
			
			
			

TABLE IX: Error examples of the top 1000 predictions for different modality configurations. The top 1000 error rates for *image*, *title*, *category* and *full* are 0.54, 0.46, 0.56 and 0.39, respectively.

task, which achieves promising performance on the fashion set popularity prediction task. We find that the combination of multimodalities and proper pooling of the instance level features, leads to the best performance. In the future, we plan to collect more data and investigate the ranking criteria in such a larger-scale problem. We also plan to extend our models to exploit generative novel fashion images.

set	orders	methods
	1 3 2 4	image
	1 2 4 3	title
	1 2 4 3	category
	1 2 4 3	full
	1 2 3 4	image
	2 1 4 3	title
	2 3 1 4	category
	1 2 4 3	full
	3 4 2 1	image
	3 4 1 2	title
	4 1 2 3	category
	4 3 1 2	full
	2 4 3 1	image
	4 2 1 3	title
	4 1 2 3	category
	4 1 2 3	full
	4 2 1 3	image
	4 1 3 2	title
	4 1 3 2	category
	4 3 1 2	full

TABLE X: Fashion item importance analysis. The first column contains the example sets, and the second column contains the orders of the item importance assigned. For each configuration, the items are sorted with by decreasingimportance.

VII. Acknowledgment

We thank Hu Hao and Eujin Rhee from Polyvore for their help in this work. This work was supported in part by New York State through the Goergen Institute for Data Science at the University of Rochester, and a FREP award from Yahoo Research.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, 2015.

[2] T. Iwata, S. Watanabe, and H. Sawada, "Fashion coordinates recommender system using photographs from fashion magazines," in *IJCAI*, 2011.

[3] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. J. Belongie, "Learning visual clothing style with heterogeneous dyadic co-occurrences," *ICCV*, 2015. [Online]. Available: http://arxiv.org/abs/1509.07473

[4] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, "Hi, magic closet, tell me what to wear!" in *ACM Multimedia*, ser. MM '12, 2012, pp. 619–628.

[5] K. Chen, K. Chen, P. Cong, W. H. Hsu, and J. Luo, "Who are the devils wearing prada in new york city?" in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 177–180.

[6] S. C. Hidayati, K.-L. Hua, W.-H. Cheng, and S.-W. Sun, "What are the fashion trends in new york?" in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 197–200.

[7] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set," in *CVPR*, 2012, pp. 3330–3337.

[8] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, "Where to buy it: Matching street clothing photos in online shops," in *ICCV*, 2015, pp. 3343–3351.

[9] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Retrieving similar styles to parse clothing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 1028–1040, 2015.

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

[11] Q. Chen, J. Huang, R. S. Feris, L. M. Brown, J. Dong, and S. Yan, "Deep domain adaptation for describing people based on fine-grained clothing attributes," in *CVPR*, 2015, pp. 5315–5324.

[12] H. Chen, A. Gallagher, and B. Girod, "Describing clothing by semantic attributes," in *European Conference on Computer Vision*. Springer, 2012, pp. 609–623.

[13] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *ICML*, 2015, pp. 720–729.

[14] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1137–1145.

[15] Y. Li, T. Mei, Y. Cong, and J. Luo, "User-curated image collections: Modeling and recommendation," in *Big Data (Big Data), 2015 IEEE International Conference on*, Oct 2015, pp. 591–600.

[16] J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua, "Inferring semantic concepts from community-contributed images and noisy tags," in *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 2009, pp. 223–232.

[17] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, A. L. Kennedy, A. Hauptmann, and J. Curtis, "Large-scale concept ontology for multimedia," *MultiMedia, IEEE*, vol. 13, no. 3, pp. 86–91, 2006.

[18] M. Wang, X.-S. Hua, J. Tang, and R. Hong, "Beyond distance measurement: constructing neighborhood similarity for video annotation," *Multimedia, IEEE Transactions on*, vol. 11, no. 3, pp. 465–476, 2009.

[19] J. Smith, L. Cao, N. Codella, M. Hill, M. Merler, Q.-B. Nguyen, E. Pring, and R. Uceda-Sosa, "Massive-scale learning of image and video semantic concepts," *IBM Journal of Research and Development*, vol. 59, no. 2/3, pp. 7–1, 2015.

[20] D. Cao, R. Ji, D. Lin, and S. Li, "A cross-media public sentiment analysis system for microblog," *Multimedia Systems*, pp. 1–8, 2014.

[21] Q. You, J. Luo, H. Jin, and J. Yang, "Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (WSDM)*, 2016, pp. 13–22.

[22] Q. You, L. Cao, Y. Cong, X. Zhang, and J. Luo, "A multifaceted approach to social multimedia-based prediction of elections," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2271–2280, 2015.

[23] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS*, 2012.

[24] M. Lin, Q. Chen, and S. Yan, "Network in network," *ICLR*, 2013.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[27] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[28] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder-decoder networks," *Multimedia, IEEE Transactions on*, vol. 17, no. 11, pp. 1875–1886, 2015.

[29] C. N. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts." in *COLING*, 2014, pp. 69–78.

[30] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *INTERSPEECH*, vol. 2, 2010, p. 3.