MMFashion: An Open-Source Toolbox for Visual Fashion Analysis

Xin Liu, Jiancheng Li, Jiaqi Wang, and Ziwei Liu

Multimedia Lab, The Chinese University of Hong Kong

Fig. 1: MMFashion, a comprehensive, flexible and user-friendly open-source toolbox. It is currently the most complete platform for visual fashion analysis in deep learning community, whose users include not only computer vision researchers, but also layman users.

Abstract—**We present MMFashion, a comprehensive, flexible and user-friendly open-source visual fashion analysis toolbox based on PyTorch. This toolbox supports a wide spectrum of fashion analysis tasks, including *Fashion Attribute Prediction*, *Fashion Recognition and Retrieval*, *Fashion Landmark Detection*, *Fashion Parsing and Segmentation* and *Fashion Compatibility and Recommendation*. It covers almost all the mainstream tasks in fashion analysis community. MMFashion has several appealing properties. Firstly, MMFashion follows the principle of modular design. The framework is decomposed into different components so that it is easily extensible for diverse customized modules. In addition, detailed documentations, demo scripts and off-the-shelf models are available, which ease the burden of layman users to leverage the recent advances in deep learning-based fashion analysis. Our proposed MMFashion is currently the most complete platform for visual fashion analysis in deep learning era, with more functionalities to be added. This toolbox and the benchmark could serve the flourishing research community by providing a flexible toolkit to deploy existing models and develop new ideas and approaches. We welcome all contributions to this still-growing efforts towards open science: https://github. com/open-mmlab/mmfashion.**

I. INTRODUCTION

Fashion plays an important role in human society. People actively engage in fashion-related activities as means of self-expressing and social connection [8], [9]. With the rapid growth of e-commerce, not only the professional researchers, but also the layman users tend to seek assistance from intelligent visual fashion analysis tools. Such enthusiasm further motivates the evolution of combining fashion with deep learning toolkit. Numerous fashion-related tasks, e.g., clothes attribute prediction, fashion recognition and retrieval, fashion parsing and segmentation, fashion compatibility analysis and recommendation, have been widely applied in online shopping platforms. These state-of-the-art algorithms are being pushed forward by worldwide researchers in a day-by-day pace. Though intelligent fashion industry is highly developed, there exists no unified platform or codebase to integrate all these diverse fashion analysis tools compared with other subfields such as object detection and video action analysis, shown as Table I. Such dilemma not only hinders the steps of professional researchers in advancing algorithms, but also discourages layman users in involving artificial intelligence to their daily life.

In order to bridge the aforementioned gaps, we propose **MMFashion**, a comprehensive, flexible and user-friendly open-source visual fashion analysis tool based on PyTorch [10]. MMFashion has several appealing main features: **1) Supporting Diverse Tasks.** We support a wide spectrum of fashion analysis tasks, which covers almost all the mainstream tasks in current research community. Together with each task, we also include popular datasets and representative methods. **2) Modular Design.** We decompose the entire framework into several re-configurable modules. With this manner, developers

TABLE I: Comparison of different high-starred fashion analysis toolkits: plenty of open-source codebases target on particular aspects of fashion analysis tasks, while none of them comprehensively support multiple tasks.

	Attribute Prediction	Cloth Retrieval	Landmark Detection	Detection and Segmentation	Compatibility and Recommendation
Fashion-MNIST [1]	✓				
tf.fashionAI [2]			✓		
fashion-detection [3]				✓	
deep-fashion-retrieval [4]		✓			
fashion-landmarks [5]			✓		
fashion-recommendationv [6]		✓			✓
polyvore [7]					✓
MMFashion	✓	✓	✓	✓	✓

can easily add or reconstruct a customized fashion analysis pipeline by combining different modules. Our provided configuration files are easy to understand even for layman users. **3) Detailed Benchmark Study.** We construct a comprehensive model zoo and measure the accuracies of individual models on a common ground. In this way, users can effortlessly download the corresponding pretrained models according to their application and accuracy requirement. At the meanwhile, professional researchers can refer to our benchmark study to evaluate their model performance.

In the following of this paper, we will first introduce the supported tasks and application scenarios, then highlight the main features of our codebase, and finally present the benchmark studies. We hope our proposed MMFashion can 1) serve as a unified platform to push forward the research efforts in fashion community, and also 2) benefit layman users to leverage the recent advances in visual fashion analysis.

II. SUPPORTED TASKS

A. Fashion Attribute Prediction

Fashion Attribute Prediction predicts the attributes of the cloth, e.g., print, T-shirt, blouse, etc. Such attribute prediction belongs to multi-label prediction problems. Following [11], the top-k recall rate is adopted as the measuring criteria.

B. Fashion Recognition and Retrieval

Fashion Recognition and Retrieval aims at determining whether two images belong to the same clothing item [3]. The two images of an image-pair can be taken from different sources. "In-shop Clothes Retrieval" indicates both images are in-shop. "Consumer-to-Shop Clothes Retrieval" means that one images is provided by the consumer and the other one is an in-shop counterpart. This task is of great use in e-commence scenarios nowadays.

C. Fashion Landmark Detection

Fashion landmarks are key-points located at the functional region of clothes, e.g., neckline, hemline and cuff [3]. The positions of landmarks defined on the fashion items are detected in this task.

D. Fashion Parsing and Segmentation

Object detection and instance segmentation are fundamental techniques in computer vision area. Fashion Parsing and Segmentation is specially designed for detection and segmentation on cloth data.
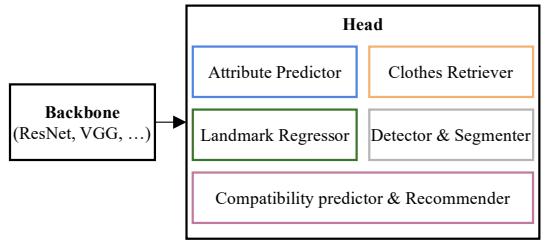

Fig. 2: Framework of MMFashion.

E. Fashion Compatibility and Recommendation

This task is composed of two main functions: 1) determining the compatibility relationship of a given fashion outfit; 2) recommending an item that matches with the existing set [7].

III. ARCHITECTURE

A. Model Representation

Although the model architectures of different tasks vary from each other, they have common components, which can be roughly summarized into the following classes.

Backbone. Backbone is the part that transforms an image to feature maps, such as a ResNet-50 [12] without the last fully connected layer.

Head. Head is the part that takes the features as input and makes task-specific predictions, such as attribute predictor, clothes retriever and landmark regressor.

With the above abstractions, the framework is illustrated in Fig. 2. We can develop our own methods by simply creating some new components and assembling existing ones.

B. Training Pipeline

We use the unified training pipeline with hooking mechanism as shown in MMDetection [13]. The training processes of these fashion analysis tasks share a similar workflow, where training epochs and validation epochs run iteratively and validation epochs are optional. In each epoch, we forward and backward the model by many iterations. To make the pipeline more flexible and easy to customize, we define a minimum pipeline which just forwards the model repeatedly. In order to run the training process of different fashion analysis tasks, we perform some self-defined operations before or after some specific steps, such as updating learning rate, saving checkpoint and evaluate the pretrained model.

TABLE II: Mean Per-Attribute Prediction performance.

Backbone	Pooling	Loss	Top-5 Recall(%)	Top-5 Accuracy(%)
VGG16	Global Pooling	Cross-Entropy	13.70	**99.81**
VGG16	Landmark Pooling	Cross-Entropy	14.79	99.27
ResNet50	Global Pooling	Cross-Entropy	23.52	99.29
ResNet50	Landmark Pooling	Cross-Entropy	**30.84**	99.30

IV. BENCHMARK STUDY

A. Dataset

DeepFashion. DeepFashion is a large-scale clothes database, which contains over 800,000 diverse fashion images ranging from well-posed shop images to unconstrained consumer photos [3]. We use its corresponding benchmarks for attribute prediction, clothes retrieval, landmark detection respectively. Additionally, we build the clothes detection and segmentation annotations as COCO-style based on "In-Shop Clothes" benchmark to serve *Fashion Parsing and Segmentation* task. From our knowledge, this human-labeled segmentation benchmark, even strict with hairs, is the most accurate open-source dataset in fashion community at present.

Polyvore. Polyvore contains rich multimodel information, e.g., images and text descriptions of fashion items, associated tags, popular score and type information, that makes it applicable for *fashion compatibility learning*. Here we use [14] reorganized version, that includes two split versions: an eaiser("Polyvore") one that allows overlapped items in train and test sets, and a difficult("Polyvore-D") one that ensures no garment appears in more than one set.

B. Evaluation Metrics

Fashion Attribute Prediction. We use the standard top-k recall rate and accuracy [11] to evaluate the prediction model. For each cloth image, we assign k highest-ranked attributes and compare the assigned attributes with the ground-truth attributes. We compute the recall rate and the accuracy for each attribute separately, shown as Equation 1:

$$per - attribute - recall = \frac{1}{c} \sum_{i=1}^{c} \frac{N_i^{tp}}{N_i^g}$$
$$per - attribute - accuracy = \frac{1}{c} \sum_{i=1}^{c} \frac{N_i^{tp} + N_i^{tn}}{N_i^p} \quad (1)$$

where c is the number of attributes, N_i^{tp} is the number of true positive predicted image, N_i^{tn} is the number of true negative predicted image for attribute i, N_i^g is the number of ground-truth attributes for attribute i, and N_i^p is the number of predictions for attribute i.

Clothes Retrieval. Similarly, we deploy the top-k recall rate to evaluate the clothes retrieval model, shown as Equation 1.

Landmark Detection. Normalized error (NE) [5], defined as $L2$ distance between the predicted landmarks and ground truth landmarks in the normalized coordinate space, is used to evaluate the fashion landmark detection, explained as Equation 2. dx, dy mean the distance between the predicted and the ground truth in x and y coordinates, w, h are the
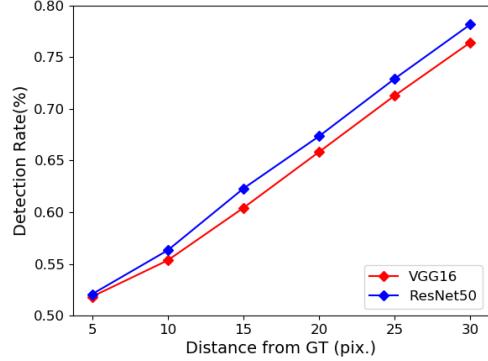

Fig. 3: Performance of fashion landmark detection on different distances from the ground truth. (px) represents pixels.

TABLE III: Fashion detection and segmentation results on different scales of images.

IoU	area	AP_{bbox}	AP_{mask}
0.50	all	0.815	0.759
0.75	all	0.663	0.637
0.50 : 0.95	all	**0.599**	**0.584**
0.50 : 0.95	small	0.118	0.071
0.50 : 0.95	medium	0.398	0.352
0.50 : 0.95	large	0.608	0.650

image width and height. Considering some landmarks are invisible due to challenging body pose, we only count the visible landmarks.

$$NE = \sqrt{(\frac{dx}{w})^2 + (\frac{dy}{h})^2} \quad (2)$$

Cloth Detection and Segmentation. We extend this function module using mmdetection codebase [13], that adopts standard evaluation metrics for COCO dataset and multiple IoU thresholds from 0.5 to 0.95 are applied. The detection and segmentation performance are measured with mean Average Precision(mAP).

Fashion Compatibility and Recommendation. Fill-in-the-Blank and Fashion-Compatibility-Prediction are utilized to evaluate the model performance, shown as Fig. IV. *Fill-in-the-Blank* gives a sequence of fashion items, and requests to select one item from four choices that is most compatible with those given items. The performance is measured as the accuracy of correctly answered questions. *Fashion Compatibility Prediction* requests to score a candidate outfit in order to determine if they are compatible or not. Performance is evaluated using the area under a receiver operating curve(AUC) [7].

C. Benchmarking Results

Fashion Attribute Prediction. We reimplement the method in [3] to predict 1000 attributes per image and report the top5 mean per-attribute recall rate and accuracy in Table II. The "landmark pooling" method is to extract and to utilize local feature maps around the landmark l, while the basic "global pooling" simply uses the global features of the whole image.

Fashion Recognition and Retrieval. We complete two subtasks here, in-shop cloth retrieval and consumer-to-shop cloth
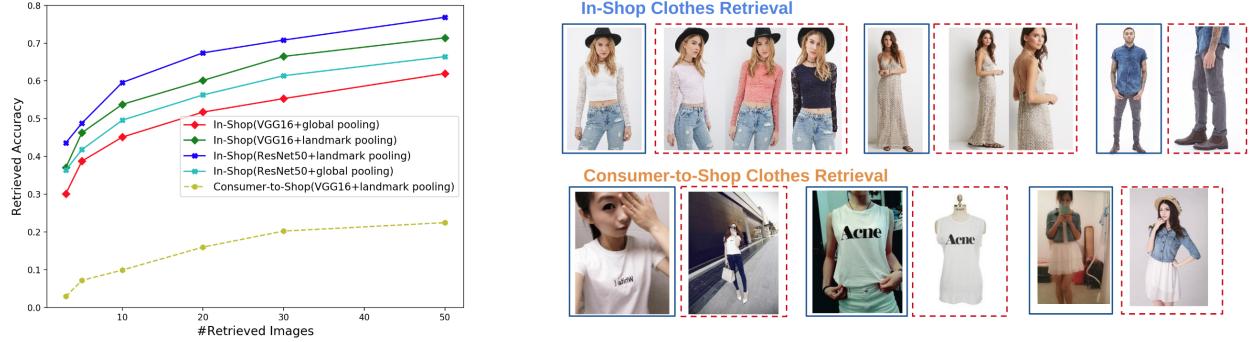
Fig. 4: Fashion Recognition and Retrieval. *Left*: Retrieval accuracies of different methods on In-shop and Consumer-to-Shop Clothes Retrieval Benchmarks. *Right*: Visual examples of cloth retrieval. Figures in blue squares are query images, in red squares are the retrieved images.

TABLE IV: *Left*: Fill-in-the-Blank(FITB) Accuracy and Fashion Compatibility Prediction AUC in two split versions of Polyvore dataset. *Right*: Visual Results of these two tasks.

Dataset	Embedding Projection	FITB Acc.(%)	Compat. AUC
Polyvore-D	fully-connected layer	50.4	0.80
Polyvore-D	learned metric	**55.6**	0.84
Polyvore	fully-connected layer	53.5	**0.85**

retrieval, shown in Fig. 4. The recall rate on consumer-to-shop task is lower than in-shop due to more challenging scenes and body poses.

Fashion Landmark Detection. The performance of landmark detection ([5], [15]) is represented as the percentage of detected landmarks (PDL), evaluating on different distances from the ground truth, shown as Fig. 3.

Fashion Parsing and Segmentation. We follow Mask-RCNN [16] to construct fashion detection and segmentation framework, using ResNet50-FPN as backbone.We measure the average precision on bounding boxes(AP_{bbox}) and on masks(AP_{mask}) in different IoU ratios and object scales. Statistic results are shown in Table III.

Fashion Compatibility and Recommendation. We use the method in [14] to complete fashion compatibility learning that develops different embedding projection strategies. Higher Fill-in-the-Blank accuracy and higher compatibility AUC means better performance, shown as Table IV.

V. IMPACTS AND FUTURE PLAN

We released MMFashion v0.1 in November 2019, while continuously maintaining and updating it. MMFashion has already received 324 stars, 21 watches and 66 forks on GitHub in less than half a year. In the future, we plan to include not only fashion recognition tasks, but also fashion synthesis tasks, such as virtual try-on. With this endeavor, MMFashion is evolving into a unified platform that eases the users to engage in intelligent fashion systems. We hope this toolkit can help join forces and push forward the development of intelligent models in fashion community.

REFERENCES

[1] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel dataset for benchmarking machine learning algorithms. 2

[2] "tf.fashionai," https://github.com/HiKapok/tf.fashionAI, 2018. 2

[3] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016. 2, 3

[4] "deep-fashion-retrieval," https://github.com/ihciah/deep-fashion-retrieval, 2018. 2

[5] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang, "Fashion landmark detection in the wild," in *ECCV*, 2016. 2, 3, 4

[6] "fashion-recommendation," https://github.com/khanhnamle1994/fashion-recommendation, 2019. 2

[7] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *ACM MM*, 2017. 2, 3

[8] X. Liu, Y. Sun, Z. Liu, and D. Lin, "Learning diverse fashion collocation by neural graph filtering," *arXiv preprint arXiv:2003.04888*, 2020. 1

[9] W. Shi, T.-W. Hui, Z. Liu, D. Lin, and C. C. Loy, "Learning to synthesize fashion textures," *arXiv preprint arXiv:1911.07472*, 2019. 1

[10] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017. 1

[11] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," *arXiv preprint arXiv:1312.4894*, 2013. 2, 3

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 2

[13] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019. 2, 3

[14] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. Forsyth, "Learning type-aware embeddings for fashion compatibility," in *ECCV*, 2018. 3, 4

[15] S. Yan, Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Unconstrained fashion landmark detection via hierarchical recurrent transformer networks," in *ACM MM*, 2017. 4

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017. 4