IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 25, 2021, accepted March 22, 2021, date of publication March 29, 2021, date of current version April 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3069245

From Street Photos to Fashion Trends: Leveraging User-Provided Noisy Labels for Fashion Understanding

FU-HSIEN HUANG[1], HSIN-MIN LU[2], (Member, IEEE), AND YAO-WEN HSU[3]
[1]Master Program in Statistics, National Taiwan University, Taipei 10617, Taiwan
[2]Department of Information Management, National Taiwan University, Taipei 10617, Taiwan
[3]Department of International Business, National Taiwan University, Taipei 10617, Taiwan

Corresponding author: Hsin-Min Lu (luim@ntu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2410-H-002-077-MY3 and Grant MOST 109-2634-F-002-045.

ABSTRACT There is increased interest in using street photos to understand fashion trends. Though street photos usually contain rich clothing information, there are several technical challenges to their analysis. First, street photos collected from social media sites often contain user-provided noisy labels, and training models using these labels may deteriorate prediction performance. Second, most existing methods predict multiple clothing attributes individually and do not consider the potential to share knowledge between related tasks. In addition to these technical challenges, most fashion image datasets created by previous studies focus on American and European fashion styles. To address these technical challenges and understand fashion trends in Asia, we created RichWear, a new street fashion dataset containing 322,198 images with various text labels for fashion analysis. This dataset, collected from an Asian social network site, focuses on street styles in Japan and other Asian areas. RichWear provides a subset of expert-verified labels in addition to user-provided noisy labels for model training and evaluation. We propose the Fashion Attributes Recognition Network (FARNet) based on the multi-task learning framework to improve fashion recognition. Instead of predicting each clothing attribute individually, FARNet predicts three types of attributes simultaneously, and, once trained, this network leverages the noisy labels and generates corrected labels based on the input images. Experimental results show that this approach significantly outperforms existing methods. Applying the trained model to the RichWear dataset, we report Asian fashion trends and street styles based on predicted labels and image clusters from latent feature vectors.

INDEX TERMS Deep learning, fashion dataset, fashion trends, image clustering, image recognition, multi-label classification, multi-task learning, noisy labels.

I. INTRODUCTION

As interest has increased in the possible relationships between artificial intelligence (AI) and fashion, more and more approaches are being proposed for fashion recognition and understanding. Meanwhile, fashion retailers are using AI technologies in inventory management, clothing recommendation, and virtual clothes fitting to improve their decision-making and competitive advantages [1]–[3].

One driving factor in the increasing popularity of fashion AI is that internet users upload and share massive numbers of photos online. These street photos on social media sites provide much-needed data for AI research. At the same time, the large-scale street images have led researchers to analyze street fashion using techniques such as deep learning [4]–[10] and natural language processing [11].

We have observed that most datasets used for street fashion research [4], [5], [12]–[15] are collected from social media sites based in the United States and Europe, and these images are mainly related to American and European street styles. Few datasets focus on Asian street styles. Moreover, fashion data collected from the internet usually contain user-provided labels that are inconsistent with the images and are referred to as noisy labels. Training deep learning models directly on

The associate editor coordinating the review of this manuscript and approving it for publication was Baozhen Yao.

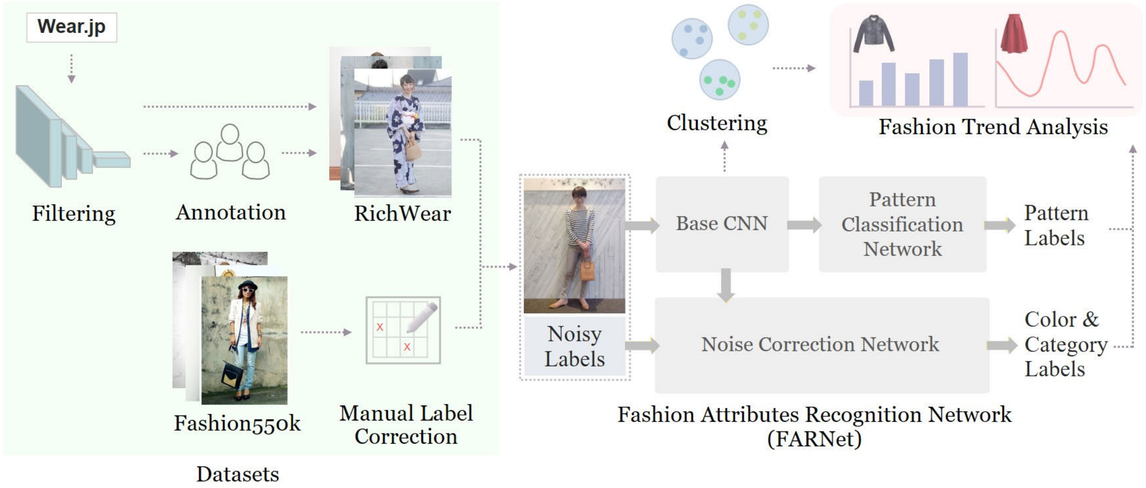


FIGURE 1. Overview of our work.

noisy labels may result in degraded predictive performance. As a result, time-consuming manual clean-ups are required when research testbeds contain these labels.

Moreover, there are rich clothing attributes in a street fashion image, such as colors, categories (e.g., shirt and pants), and patterns. Given that fine-grained attribute recognition is challenging, most previous studies built separate models to predict each clothing attribute. In this process, each model learns a classification task individually, which does not allow for the potential to share knowledge between related tasks.

To address these issues, we created RichWear, a new fashion dataset containing 322,198 high-quality street fashion images with noisy labels from the Asian fashion website WEAR.[1] The dataset focuses on the street styles in Japan and other Asian areas from 2017 to 2019. We manually annotated 4,368 images with verified clothing labels for model training. To increase the sample size, we combined the dataset with Fashion550k [5],[2] a public street fashion dataset. We also manually corrected any errors and inconsistencies in the verified labels of Fashion550k. To improve fashion recognition, we propose the Fashion Attributes Recognition Network (FARNet) that includes a Noise Correction Network and a Pattern Classification Network on top of a Convolutional Neural Network (CNN) image feature extractor. The two main networks of FARNet are trained jointly to simultaneously predict clothing colors, categories, and patterns based on the input image and its noisy labels. Also, because we are interested in undiscovered street fashion trends in Asia, aggregation of predicted labels and image clusters in the RichWear dataset allows for identification of meaningful trends and discovery of style dynamics. The overview of our work is shown in Fig. 1.

The remainder of this work is organized as follows: We discuss related studies in Section II and then present our new dataset in Section III. Section IV introduces the proposed method for fashion recognition and is followed by experimental results in Section V. Finally, we present Asian street fashion trend analysis in Section VI and conclude our study in Section VII.

II. LITERATURE REVIEW

There is a large amount of literature that develops deep learning and computer vision approaches to image understanding and fashion recognition. Our literature review focuses on three streams of related studies: (1) fashion attribute recognition, (2) street fashion analysis, and (3) street fashion datasets.

A. FASHION ATTRIBUTE RECOGNITION

Fashion attribute recognition aims to identify one or more fashion-related attributes according to input images. Recent works have shown growing interest in training machines for effective visual recognition on large-scale image datasets. Existing research on fashion attribute recognition covers the tasks of image parsing [13], [16], image classification [17]–[22], and classification with noisy labels [5], [23].

Image parsing or image segmentation is the task of segmenting and identifying all of the objects in an image. Early research used image parsing algorithms to produce pixelwise annotation for clothing attribute recognition by assigning a semantic label to each pixel in the image. Yamaguchi *et al.* [13] introduced a retrieval-based approach that retrieves images similar to the query image from a small set of hand-parsed images with labels and then uses the retrieved images to parse the clothing attributes of the query. Yang *et al.* [16] developed a clothing co-parsing system to jointly parse a set of clothing images and produce pixelwise annotation of attributes. This co-parsing system outperforms the approach of Yamaguchi *et al.* [13] and other image parsing methods of attribute recognition on the fashion image datasets.

In recent years, image classification has been widely used for attribute recognition with good performance. The goal

[1] https://wear.jp/.
[2] https://esslab.jp/~ess/en/data/fashion550k/.

of image classification is to predict a single label or a set of labels for a given image. Image classification models often employ CNNs to extract features related to shapes and textures in an image and to generate predictions of relevant attributes [17]. Similar ideas have been extended to clothing recommendation [18] and fashion image retrieval [19], [20]. To improve the performance of classification, several studies introduced multi-task learning (MTL) into their methods [21], [22]. By learning all tasks jointly, MTL can significantly improve the performance of fashion recognition through sharing knowledge and leveraging information between different tasks.

While fashion images can be easily collected from social media sites, user-generated fashion data often contain error-prone labels that are not necessarily consistent with the images. Moreover, human annotation of image labels is often expensive and time-consuming, and training on noisy labels directly results in a substantial decrease in the performance of deep neural networks [24]. Therefore, understanding how to leverage noisy labels and improve predictive performance is a critical research direction. Veit *et al.* [23] proposed a multi-task network that jointly learns to clean noisy labels and to perform multi-label classification. Their results show that this approach is better than directly training an image classifier on noisy labels. Following the work of Veit *et al.* [23], Inoue *et al.* [5] developed a multi-label classification model for a fashion dataset that includes a large number of instances with noisy labels and a small set of instances with verified labels. By leveraging noisy labels and cleaning them for model training, this method both reduces the overfitting problem that can occur when training is limited to a small amount of label-verified data and improves overall model performance.

B. STREET FASHION ANALYSIS

Street fashion does not originate from studios or runways, but from real-life streetwear [25]. As more internet users share street photos on social media, street fashion has gradually become a driving force for fashion change and fashion design [4]. Nevertheless, unlike online shopping images and runway images with standing models and simple backgrounds, street fashion images usually contain people in different poses with various backgrounds, a difference that increases the difficulty of image recognition. Street photos collected from social media sites also often have noisy labels that demand a new technical approach. In this subsection, we provide a review of studies that focus on street fashion analysis, including fashion trends discovery [4], [8], [9], style discovery and construction [7], [10], [11], [26], and popularity prediction [6], [27].

Since street fashion trends are strongly influenced by climate and culture, they may vary over time and by location. Social events like festivals and sporting matches are also factors that affect streetwear. To identify street fashion trends in a large collection of image data, several studies use not only the aforementioned attribute recognition methods but also image clustering techniques [4], [8]. Image clustering utilizes extracted image features and provides an unsupervised method for visual understanding. To detect social events that affect street fashion, Mall *et al.* [9] followed the work of Matzen *et al.* [8] by building separate CNNs for each clothing attribute recognition and performing image clustering. They also developed a parametric model to discover long-term trends and identify short-term spikes caused by social events. However, some models used to generate image features for clustering were trained only with noisy labels, without verified labels as the ground truth [4], [26].

Specific collocations of garments create different fashion styles, and these styles depend on specific visual features, such as color theme and collar shape [7]. Several studies distinguish between different street fashion styles (e.g., casual and rock) in image data by using CNN models [10], [26] or the polylingual Latent Dirichlet Allocation (PolyLDA) model [11]. Given that certain clothing features make up unique fashion styles, Ma *et al.* [7] first proposed a multimodal deep learning model to construct fashion styles across brands and over time.

Several studies have extended street fashion analysis to popularity prediction [6], [27]. By utilizing likeability (the number of likes or votes), these studies predicted the fine-grained popularity of an outfit or look. Yamaguchi *et al.* [27] applied the image parsing method based on [13] to recognize clothing attributes as image content factors. They then employed linear regression models to predict the popularity of street photos using content factors and social factors. Lo *et al.* [6] developed a deep temporal sequence learning framework to predict the popularity of individual outfits.

C. STREET FASHION DATASETS

Several street fashion studies created new datasets to facilitate model training and evaluation. These datasets each contain a different number of street fashion images with text labels or other types of annotations dependent on the original research purposes. The most commonly-used text labels are related to clothing attributes, such as category, color, and sleeve length, or to fashion styles, such as ethnic, casual, and fairy. Table 1 lists selected public street fashion datasets created by previous studies. We exclude some large fashion datasets, like the DeepFashion dataset [22] and the Runway dataset [28], that mainly contain online shopping or runway images rather than street fashion.

Most studies collected street photos from social media sites, such as Instagram and Chictopia [4], [5], [8], [12]–[14]. A few studies utilized search engines such as Google to gather street photos by using fashion-related words as queries [10], [29]. We find that most datasets cover only one or two clothing attributes, such as clothing category and color. The only exception is STREETSTYLE-27K [8], which incorporates 12 clothing attributes. However, each of its images contains only the human head and torso, instead of the full body.

TABLE 1. Comparison of street fashion datasets.

Dataset [a]	Data Source	Number of Images	Type of Labels	Main Source by Geographic Area
Fashionista [14]	Chictopia	158,235 (Full body)	Clothing category and body pose (685 images with verified labels)	United States and Europe
Paper Doll [13]	Chictopia	339,797 (Full body)	Clothing category and body pose (Noisy labels)	United States and Europe
HipsterWars [29]	Google	1,893 (Full body)	5 fashion styles (Verified labels)	Various
Fashion144k [12]	Chictopia	144,169 (Full body)	Clothing category and color (Noisy labels)	United States and Europe
STREETSTYLE-27K [8]	Instagram	14.5 million (Head and torso)	12 clothing attributes (Noisy labels & a 27,000-image subset with verified labels)	44 cities around the world
FashionStyle14 [10]	Search engine	13,126 (Full body)	14 fashion styles (Verified labels)	Various
Fashion550k [5]	Chictopia	407,772 (Full body)	Clothing category and color (Noisy labels & a 5,300-image subset with verified labels)	United States and Europe
Street Fashion Style [4]	Chictopia	293,105 (Full body)	Clothing category and 15 fashion styles (Noisy labels)	United States and Europe
ModaNet [15]	Paper Doll	55,176 (Full body)	Bounding boxes around 13 clothing categories	United States and Europe

[a] Datasets that primarily consist of runway or online shopping images are excluded.

In addition, most street fashion datasets [4], [5], [12]–[15] primarily contain images from the United States and Europe.

D. RESEARCH GAPS

After reviewing previous research, we have identified several gaps that merit further investigation. First, some prior studies directly trained models with noisy labels, which may have deteriorated predictive performance. Developing more effective methods may address the problems associated with noisy labels and improve image classification. Second, most works built models to separately classify each type of clothing attribute, which did not allow for knowledge sharing between related tasks. Third, few street fashion datasets cover more than one clothing attribute with verified labels. The limited type of attributes cannot fully illustrate the rich information inherent in street fashion photos. In addition, most datasets primarily contain photos that originate in the United States or Europe. Few focus on street fashion images from Asia. Finally, few papers have investigated fashion trends in Asia. As Asian consumer spending rises, many fashion companies have been deepening their understanding of Asian fashion trends in order to meet consumers' needs [30]. Given the growing importance of the Asian fashion market, it is valuable to explore this region's street fashion trends and to identify its style dynamics.

III. THE RichWear DATASET

The goal of our study is to create a deep learning architecture in order to understand individual street photos and analyze the aggregated fashion trends in Asia. However, existing street fashion datasets mainly cover American and European street photos, which are not suitable for our goal. To address this issue, we have created RichWear,[3] a new dataset that focuses on the street fashion styles in Japan and other Asian areas.

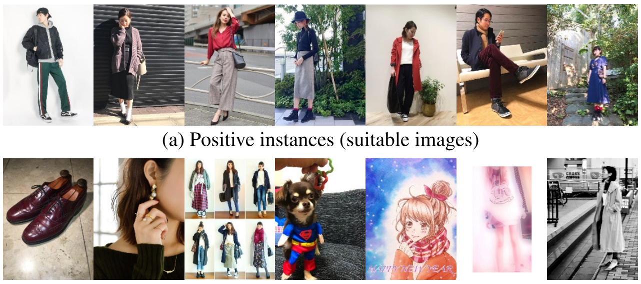
A. IMAGE COLLECTION AND CLEANING

We collected street fashion images together with their text labels from WEAR,[4] a popular fashion coordination website in Japan. We crawled 389,633 images with upload date, users'

[3] RichWear is openly available at https://github.com/hsinmin/richwear.
[4] https://wear.jp/.

gender and country, user-provided clothing labels, clothing brands, and user-created hashtags. Our dataset covers photos from 2017 to 2019. The numbers of images of each season are roughly the same, which is a plus for tracking fashion trends.

Many images uploaded by users are not suitable for model training. For example, images of animals and body parts are bad sources for street fashion understanding. Blurred images and those that are heavily distorted by photo filters must also be excluded. Following Simo-Serra and Ishikawa [26], we created a CNN-based image filter to remove unsuitable images. We formulated this task as a binary classification problem and manually annotated a subset for training and testing. Fig. 2 shows examples of positive and negative instances. The positive instances have a fully-visible individual in the photos; the negative instances are photos that contain a single category, body parts, individuals, animals, illustrations, and distorted images. Since our task includes color prediction, black-and-white images are also considered as negative instances. To train and evaluate this image filtering task, we annotated 5,850 images and then split them into 3,500 training images (positive: 1,509, negative: 1,991), 350 validation images (positive: 168, negative: 182), and 2,000 test images (positive: 850, negative: 1,150).


(a) Positive instances (suitable images)

(b) Negative instances (unsuitable images)

FIGURE 2. Examples of positive and negative instances (suitable and unsuitable images).

To construct the image filter, we used the training set to fine-tune a VGG16 model [31] pretrained on the ImageNet dataset. We adopted the stochastic gradient descent (SGD) optimizer with a learning rate of 0.00005 for parameter learning and also applied data augmentation for fine-tuning. The validation set was used for choosing the best model. Our final model achieved an accuracy of 89.75% on the test set. This image filter can effectively filter out most of the unsuitable images with a precision of 91.48%, a recall of 90.61%, and an F1-score of 91.04%. After applying this model to all crawled images, we filtered out improper images and retained 322,198 images. Among those retained, 64.2% are female, and 35.8% are male. Although the number of female images is about two times the number of male ones, we do have a sufficient number of images to analyze street fashion by gender. Also, 89.1% of the users in RichWear are from Japan, and more than 90% are from Asia, providing us a good data source from which to understand street styles in Asia.

B. LABEL PREPROCESSING AND ANNOTATION

The user-provided clothing labels include pairs of category and color in the form of *category (color)*, such as *Pants (Black)* and *Jacket (White)*. We split the *category (color)* pairing into *category* and *color* and combined similar categories. For example, one-piece dress, shirt dress, pinafore dress, and tunic are combined into the *dress* category. We also removed some accessories, such as earrings and hats. We kept a total of 36 classes that include the 12 colors and 24 categories that are shown in Table 2.

TABLE 2. Clothing attributes in RichWear.

Attribute	Class			
Color	Black	Gray	White	Beige
	Orange	Pink	Red	Green
	Brown	Blue	Yellow	Purple
Category	Top	T-Shirt	Shirt	Cardigan
	Blazer	Sweatshirt	Vest	Jacket
	Dress	Coat	Skirt	Pants
	Jeans	Jumpsuit	Kimono_Yukata	Swimwear
	Stockings	Shoes	Sandals	Boots
	Pumps	Sneakers	Scarf	Bag
Pattern	Solid	Striped	Floral	Plaid
	Spotted			

As discussed previously, user-provided clothing labels are noisy and require manual verification. To create a subset of images that contain verified labels, we instructed six experts who live in Asia and have sufficient knowledge of Asian fashion to manually annotate 4,368 images with correct color and category labels. All images that went through manual annotation were checked by the first author to ensure that they are photos of fully-visible individuals suitable for model training.

In addition, clothing pattern recognition can facilitate fashion understanding. Therefore, we asked the annotators to additionally annotate each of the 4,368 images with one of the five pattern classes shown in Table 2. We trained the annotators by showing them examples of images with different clothing colors, categories, and patterns. We also asked the annotators to practice annotating new images to ensure their understanding of the annotation rules. After they finished their tasks, the first author manually verified these labels for quality assurance. The total annotation process took more than six weeks. After manual annotation, each image in the verified subset contains both noisy and human-verified labels. We split the verified subset into 3,043 training images, 325 validation images, and 1,000 test images.

Fig. 3 shows the quality of the noisy labels for the verified subset. The rate of labels positively verified is the proportion of images with user-labeled classes that are verified by annotators to actually belong to these classes. The rate of label coverage is the proportion of images with verified classes that were labeled as such by users. The quality of the noisy labels

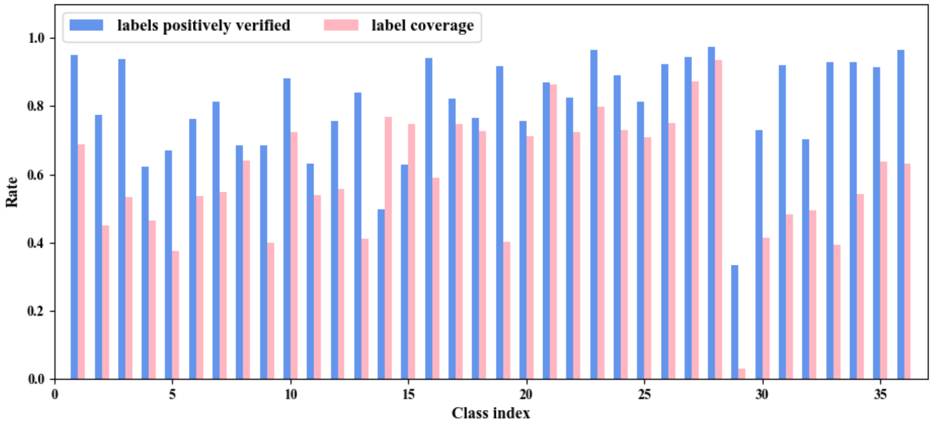


FIGURE 3. Quality of noisy labels. The class index contains 36 classes in the following order: black, gray, white, beige, orange, pink, red, green, brown, blue, yellow, purple, top, t-shirt, shirt, cardigan, blazer, sweatshirt, vest, jacket, dress, coat, skirt, pants, jeans, jumpsuit, kimono_yukata, swimwear, stockings, shoes, sandals, boots, pumps, sneakers, scarf, and bag.

differs to some extent based on different classes, and the small objects (e.g., stockings and shoes) have an especially low rate of labels positively verified and of label coverage because most users tend to mislabel or ignore them. Due to the noise in every class, it is inappropriate to directly use the noisy labels for further analysis without correction.

C. DATASET COMBINATION

To increase training size for better recognition performance, we combined the training set of RichWear with a verified subset of the Fashion550k dataset [5]. Fashion550k, which contains 407,772 street fashion images with clothing attributes in colors and categories complementary to RichWear, provides additional data for our tasks. Fig. 4 shows examples of images from RichWear and Fashion550k. Our initial inspection suggests that there are inconsistencies and errors in the 5,300 images with verified labels in Fashion550k. As shown in Fig. 5, some images contain the same clothing category but have inconsistent labels, and some images have labels that are inconsistent with their photos. To address this issue, a human annotator spent four weeks manually correcting the verified labels. We removed several accessory classes and merged the original 66 into 36 classes. Likewise, we annotated each image with one of the five pattern classes. This process gave us 4,307 corrected images from Fashion550k. We combined these images with the training set of RichWear to create an augmented training set containing 7,350 images and then used the augmented training set in the subsequent model training process. We discuss the effect of dataset combination on recognition performance at the end of Section V.

IV. PROPOSED METHOD

We propose the Fashion Attributes Recognition Network (FARNet) to simultaneously recognize three types of clothing attributes, including colors, categories, and patterns, in noisy-labeled images. FARNet contains two main components built on top of a CNN image feature extractor. The first component is a Noise Correction Network extended based on the model of Inoue *et al.* [5]. This network corrects noisy labels for images and generates corrected multi-labels of clothing colors and categories. The second component is a Pattern Classification Network that classifies each image into one of the five clothing patterns. We trained FARNet using the MTL framework. The benefit of MTL compared with single-task learning (STL) is that it allows for exploration of latent connections and facilitates knowledge sharing between tasks, leading to an overall improvement of model performance [21]. We describe our model architecture and the loss functions in this section.

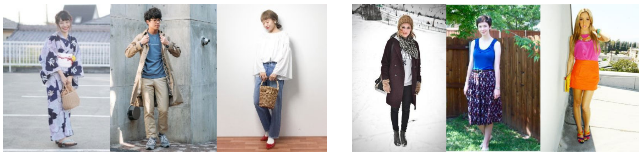

(a) RichWear dataset
(b) Fashion550k dataset

FIGURE 4. Examples of images from RichWear and Fashion550k.

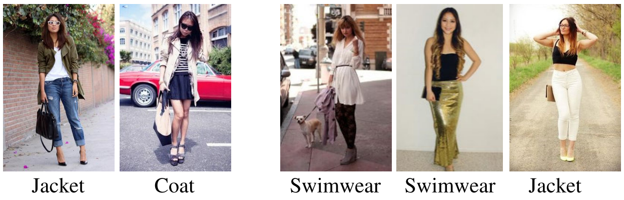


FIGURE 5. Examples of inconsistency (left) and errors (right) in the verified labels of Fashion550k.

A. MODEL ARCHITECTURE

Fig. 6 presents the overall architecture of the proposed FARNet, which contains a Noise Correction Network g and a Pattern Classification Network h that are built on top of a base CNN f. For each street fashion image I, there are a set of noisy labels y, a set of human-verified labels v, and

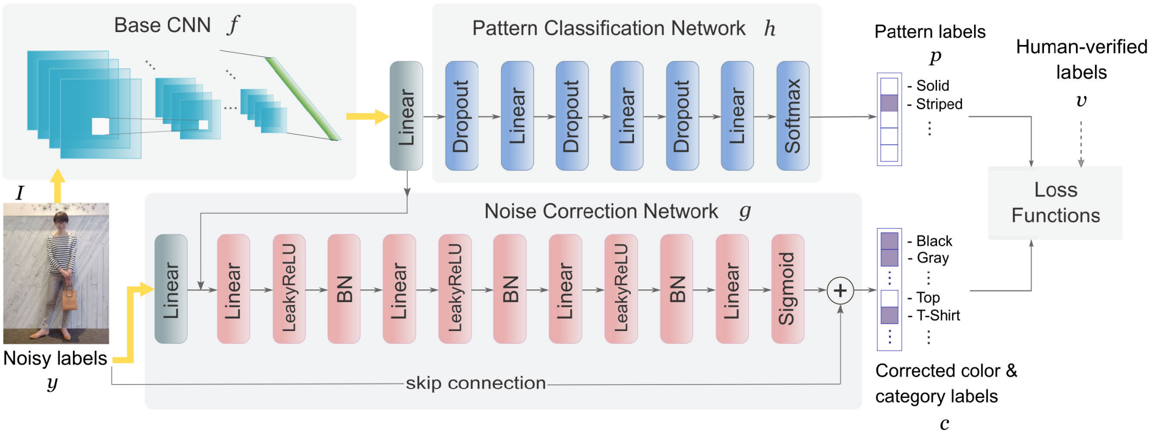


FIGURE 6. The overall architecture of FARNet.

corresponding image features $f(I)$ that are extracted from f. The noisy label vector and the verified label vector are sparse binary vectors. The two inputs of our networks, $f(I)$ and y, are separately projected into a 512-dimensional embedding and a 16-dimensional embedding. The network g is trained to output the 36-dimensional corrected label vector c that represents the predicted attributes of clothing colors and categories, and the network h learns to predict the five-dimensional clothing pattern vector p for the image I.

1) THE NOISE CORRECTION NETWORK

The Noise Correction Network g learns a structure by mapping the noisy labels y to the human-verified labels v conditional on visual information $f(I)$. The human-verified labels v are used as the ground truth to supervise the Noise Correction Network. The input of the Noise Correction Network is the concatenation of the low-dimensional embeddings of $f(I)$ and y. This network contains three linear hidden layers of 512 units each and one output layer that is followed by the sigmoid function. We add a leaky rectified linear unit (LeakyReLU) layer and a batch normalization (BN) layer [32] after each linear hidden layer. LeakyReLU and BN can address the saturation problem and the vanishing gradients during training. BN also regularizes the network for avoiding overfitting and removes the need for a dropout layer [33]. Here we use a skip connection that adds the noisy labels y to the output values as the final output. To obtain valid corrected labels c, we clip the final output to an interval of [0, 1]. If g' is the residual module of g before adding y to the output values, the Noise Correction Network can be denoted as:

$$c = \max(\min(g'(f(I), y) + y, 1), 0). \qquad (1)$$

Although the Noise Correction Network is similar to the label cleaning network of Inoue *et al.* [5], there are several noticeable differences between the two models. First, our Noise Correction Network has three hidden layers and is deeper than the original label cleaning network with one hidden layer. Because deeper neural networks can learn higher-level features [34], our network benefits from the deeper architecture. Second, we have adopted a LeakyReLU layer and a BN layer after each hidden layer to reduce overfitting. By contrast, Inoue *et al.* [5] used rectified linear unit (ReLU) and BN after the hidden layer. The ReLU layer performs a threshold operation. However, the model learning process may lead to nodes that always receive negative inputs and thus never activate. These nodes contribute to neither the gradient nor the output and are often referred to as "dead." This is known as the dying ReLU problem [35]. To avoid this problem, we use LeakyReLU, which also performs a threshold operation but additionally allows a small, non-zero gradient for negative inputs. The weights of those nodes that are not active with ReLU are adjusted in model training. Third, we have added a sigmoid activation function at the output layer. After applying the sigmoid, we convert the output of the Noise Correction Network into a value between 0 and 1 before adding y, which leads to better prediction performance.

2) THE PATTERN CLASSIFICATION NETWORK

FARNet also contains a Pattern Classification Network h, which utilizes visual information $f(I)$ to predict the clothing pattern p for a street fashion image. The human-verified labels v are used as the ground truth to supervise the Pattern Classification Network. The low-dimensional embedding of $f(I)$ is the only input of this network that consists of two linear hidden layers of 256 units each and one output layer. A softmax function is applied to the output layer to normalize the outputs as predicted probabilities for the pattern classes.

To prevent overfitting, we add a dropout layer [33] after each linear layer. Although both BN and dropout can regularize the network and reduce overfitting, combining them together often results in a worse predictive performance than if they had been used separately [36], and thus each main component is equipped with its own appropriate regularizer for the best performance. Finally, the predicted pattern labels p can be utilized for subsequent fashion trend analysis. The Pattern Classification Network can be expressed as:

$$p = h(f(I)). \tag{2}$$

B. LOSS FUNCTIONS

FARNet is designed to jointly learn the label noise correction task and the pattern classification task in a MTL framework. To train the FARNet, we jointly optimize the correction loss of the Noise Correction Network and the classification loss of the Pattern Classification Network. The total loss function is expressed as:

$$\text{Loss}_T = \lambda\text{Loss}_C + (1-\lambda)\text{Loss}_P, \tag{3}$$

where Loss_C is the correction loss, Loss_P is the classification loss, and λ is the weight to control the trade-off between the two loss functions.

1) THE CORRECTION LOSS

The correction loss for the Noise Correction Network is the binary cross-entropy loss, which can be defined as:

$$\text{Loss}_C = -\sum_{i=1}^{N}\sum_{j=1}^{S_1} v_{i,j}\cdot\log(c_{i,j}) + (1-v_{i,j})\cdot\log(1-c_{i,j}), \tag{4}$$

where N is the total number of training instances, S_1 is the total number of color and category classes, $c_{i,j}$ is the predicted probability of a class j, and $v_{i,j}$ is the verified label (the ground truth) of that class. The Noise Correction Network is supervised by the verified labels of clothing colors and categories.

We have chosen the binary cross-entropy loss rather than the L_1-distance loss utilized by Inoue *et al.* [5]. The L_1-distance loss between the predicted probability $c_{i,j}$ and the verified label $v_{i,j}$ is defined as:

$$L_1\_\text{distance} = \sum_{i=1}^{N}\sum_{j=1}^{S_1} |c_{i,j} - v_{i,j}|. \tag{5}$$

Compared to L_1-distance, binary cross-entropy is able to enlarge the loss more as c diverges from v, which can give incorrect predictions more penalty. Therefore, we believe that the binary cross-entropy is more suitable for our Noise Correction Network.

2) THE CLASSIFICATION LOSS

The Pattern Classification Network uses categorical cross-entropy loss:

$$\text{Loss}_P = -\sum_{i=1}^{N}\sum_{j=1}^{S_2} v_{i,j}\cdot\log(p_{i,j}), \tag{6}$$

where N is the total number of training instances, S_2 is the total number of pattern classes, $p_{i,j}$ is the predicted probability of a pattern class j, and $v_{i,j}$ is the verified label (the ground truth) of that class. The Pattern Classification Network is supervised by the verified labels of clothing patterns.

V. EXPERIMENTS

We trained our FARNet on the 7,350-image training set that is the combination of the verified training set of RichWear and manually-corrected samples of Fashion550k. The validation set and the test set contain 325 images and 1,000 images, respectively, from the verified subset of RichWear. The remaining parts of the RichWear dataset were used for the street fashion trend analysis. In addition to the MTL framework, we also separately trained our Noise Correction Network and Patten Classification Network in STL to evaluate their performance. In this section, we present the metrics used for performance evaluation. We then introduce several baseline methods for comparison with our model, followed by the experimental details. Finally, we report and discuss the experimental results.

A. EVALUATION METRICS

Following other related studies [5], [23], [37], we adopt the mean average precision (mAP) as the evaluation metric for multi-label color and category predictions. We obtain mAP by taking the average over all average-precision for each class (AP$_{cl}$). The definition of AP$_{cl}$ follows the widely-adopted standard defined in the PASCAL Visual Objects Classes (VOC) Challenge [38], [39]. Given one class, the AP$_{cl}$ summarizes the shape of the precision-recall curve that is derived from the ranked outputs (i.e., predicted probabilities) of a model in descending order. The AP$_{cl}$ is expressed as:

$$AP_{cl} = \sum_{k\in\{1,2,\ldots N\}} Prec_{k,r}\cdot\Delta Rec_k, \tag{7}$$

where N is the total number of predictions, and k is the retrieved rank, from 1 to N, retrieving from the largest output. At each rank (k), there are corresponding precision and recall values. Rec_k is recall at rank k, and ΔRec_k is the difference between Rec_k and Rec_{k-1}. $Prec_{k,r}$ is precision at rank k, and the maximum precision of all recall values $\geq r$ is used for a level of recall r to ensure the precision-recall curve decreases monotonically. Then mAP can be formulated as:

$$mAP = \frac{1}{S_1}\sum_{cl=1}^{S_1} AP_{cl}, \tag{8}$$

where S_1 is defined as in Equation (4). For clothing pattern prediction, we choose accuracy to evaluate the performance of single-label classification. Accuracy is the fraction of predictions that a model predicts correctly.

B. BASELINES

We compare FARNet with several baseline methods, including that proposed by Inoue *et al.* [5]. We focus on comparing our Noise Correction Network with these baseline methods, the details of which are described as follows:

1) **Inoue *et al.***: We compare our model with the label cleaning network of Inoue *et al.* [5] that is similar to

TABLE 3. Performance of baselines and our methods.

Method	Colors and categories mAP (%)	Pattern Accuracy (%)
Baselines		
Inoue *et al.*	59.79	-
Noisy labels as predictions	55.73	-
Noisy labels as target	55.46	-
No visual information	57.63	-
Ours		
The Noise Correction Network (STL) [a]	66.98	-
The Pattern Classification Network (STL)	-	**79.9**
FARNet	**71.33**	79.8

[a] STL is single-task learning.

our Noise Correction Network. This baseline uses the L_1-distance loss in Equation (5).

2) **Noisy labels as predictions**: This baseline directly takes the noisy labels of the test set as predictions for performance evaluation. There are no models trained in this baseline. This naive approach indicates the usefulness of the noisy labels.
3) **Noisy labels as target**: This method uses the noisy labels of the training set to supervise the base CNN and the Noise Correction Network. That is, we use the noisy labels as the ground truth to train our model and then apply the trained model to the test set to evaluate the performance. The result of this baseline can be interpreted as the best case of training directly on the noisy labels.
4) **No visual information**: In this baseline, the only inputs to the Noise Correction Network are the noisy labels without any image features. This baseline provides noise correction performance without the help of visual information. The structure used here is identical to the Noise Correction Network except for fewer neurons in the linear hidden layers.

C. EXPERIMENTAL DETAILS

We have chosen ResNet50 [34] as the architecture for the base CNN because this residual network is easy to optimize and able to gain accuracy from its deep architecture. To obtain better quality of image features for our tasks, we use the training set to fine-tune ResNet50, which was pretrained on the ImageNet dataset. Then we employ the base CNN to extract the 2,048-dimensional feature vectors for all images in our dataset. In the fine-tuning process, data augmentation, including horizontal flipping and random cropping, is applied to the training set. We also apply center cropping to the validation set and the test set. All images we use here are resized to 256 × 256 pixels and then cropped to 224 × 224 pixels.

To overcome the potential overfitting problem during model training, we use the validation set to perform early stopping and to find the best model. We use RMSprop optimizer with a learning rate of 0.00001 for fine-tuning the base CNN and then Adam with a learning rate of 0.00005 for training the two main networks of FARNet. We also apply a learning rate decay of 0.5 after 20 epochs with no improvement in the training process. The hyperparameter λ in the FARNet loss function is 0.5.

D. EXPERIMENTAL RESULTS

We trained each baseline separately following the training process of our model. Note that all baselines are STL methods. Table 3 summarizes the performance of the baselines and our methods. Among all STL methods, our Noise Correction Network (STL) achieves the best performance on noise correction for color and category labels. Notably, our Noise Correction Network (STL) outperforms the label cleaning network of Inoue *et al.* by 7.19 percentage points. Moreover, the ''noisy labels as target'' method has the worst performance among all baselines. This means that using noisy labels as the ground truth to train the model is inappropriate. The Noise Correction Network (STL) significantly outperforms the ''no visual information'' method, which implies that label noise correction greatly benefits from the help of visual information. Finally, we compare the performance within our methods and find that there is not a large difference in the performance of pattern classification between STL (79.9%) and MTL (79.8%), but MTL indeed improves the performance on noisy label correction from 66.98% to 71.33%. Through this experiment, we demonstrate that MTL can noticeably improve the generalization performance of fashion attribute recognition tasks.

To further analyze our design, we compare the performance among different model structures in Table 4. The ''original'' model is our Noise Correction Network, and the ''using ReLU'' model is the first variant. Following the

TABLE 4. Comparison of performance between different model structures.

	Inoue *et al.*	The Noise Correction Network (STL)			
		Original	Using ReLU	Without sigmoid	Using L_1-distance
mAP (%)	59.79	**66.98**	65.53	44.73	65.32

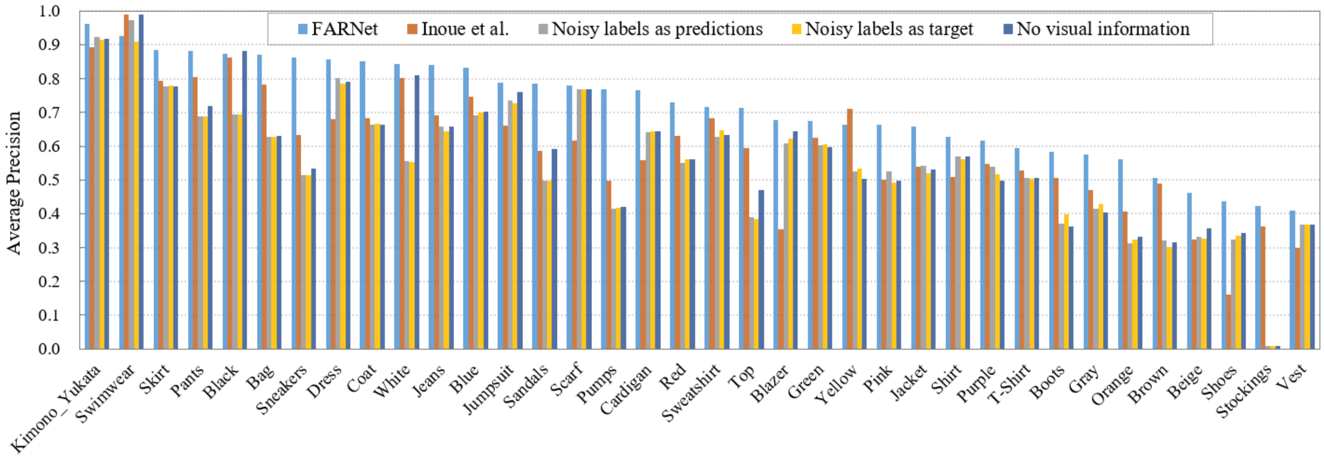


FIGURE 7. Comparison of AP$_{cl}$ for clothing color and category attributes.

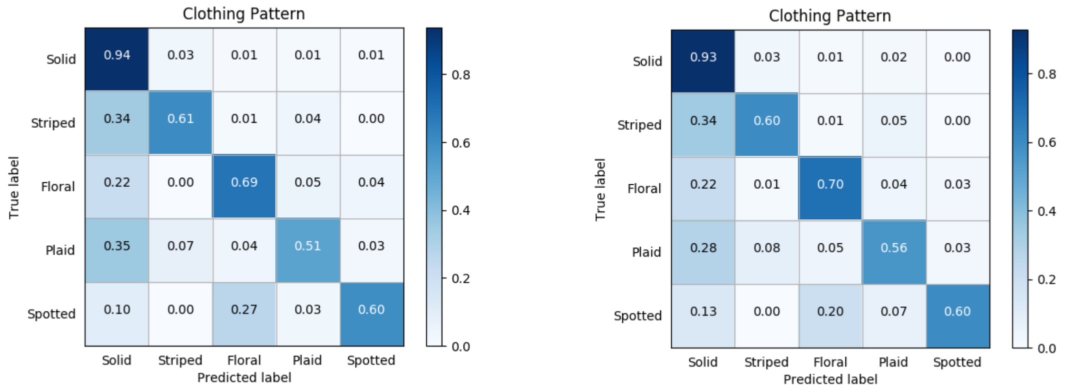
example set by Inoue *et al.* [5], it uses ReLU after each linear hidden layer instead of LeakyReLU. The second variant is the ‘‘without sigmoid’’ model that omits the sigmoid function at the output layer as in the model proposed by Inoue *et al.* [5]. In the third variant, the original binary cross-entropy loss is replaced by the L_1-distance loss of Inoue *et al.* [5]. We show that using ReLU after each linear hidden layer hurts the performance of our model. Also, omission of the sigmoid function at the output layer sharply reduces mAP of our model; applying sigmoid significantly improves the noise correction performance by 22.25 percentage points. Finally, if we replace the binary cross-entropy loss with the L_1-distance loss, the model performance deteriorates.

We present the comparison of AP$_{cl}$ for clothing color and category attributes in Fig. 7, which is sorted in descending order of AP$_{cl}$ of FARNet. Obviously, FARNet surpasses all compared baseline methods in most classes. Moreover, three-fourths of the classes achieve AP$_{cl}$ over 60% when FARNet is used. This reveals that our proposed method can correct noisy labels and predict the color and category attributes well. As for the performance difference between classes, there are at least three potential influence factors. The first factor is the number of training instances in the class. The classes with larger numbers of training instances on average have a higher AP$_{cl}$, as demonstrated in Section C of the supplemental material. The second factor is the noisy label quality, and we report its influence in the next subsection. The last factor is the visual appearance (e.g., size and shape) of the class. The classes with smaller visual appearances usually have worse recognition performance because of little filtered information at the very top layers of the model [40], [41]. For example, *stockings* on average occupies 1.89% of visible pixels in the training images, compared to 7.09% of visible pixels for *jeans* and 4.28% for *cardigan*, and the model performance of *stockings* is inferior to that of *jeans* and *cardigan*.

Fig. 8 shows the similar confusion matrices for pattern prediction of MTL and STL. We observe that the MTL method can predict *plaid* slightly better than the STL method. However, the images that contain small patterned objects tend to be classified as *solid* no matter what pattern is shown. For example, an image that contains a bag with a plaid pattern is usually classified as *solid*.

E. NOISY LABEL QUALITY AND DATASET COMBINATION

We further discuss whether the quality of the noisy labels and combined datasets (i.e., RichWear + Fashion550k) affect the model performance. To investigate the effect of the quality of the noisy labels on the performance, we adopt a method similar to [23]. We first group the color and category classes into six equally-sized groups that are ranked from very noisy to very clean based on the average of the rate of labels positively verified and the rate of label coverage. We then compute the mAP improvement over the third baseline (i.e., noisy labels as target) for each quality group. We provide the performance improvement for all classes in Section C of the supplemental material. Fig. 9 shows the mAP improvement with respect to the noisy labels' quality. FARNet has effective noise correction and performance improvement for all levels of noisy label quality. Among the six quality groups,



(a) Single-task learning

(b) Multi-task learning

FIGURE 8. Confusion matrices for clothing pattern prediction.

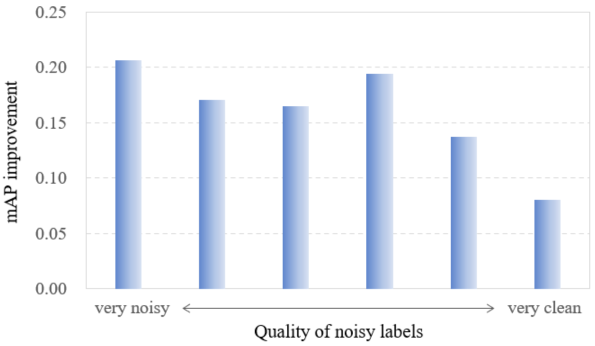


FIGURE 9. Performance improvement with respect to the quality of the noisy labels.

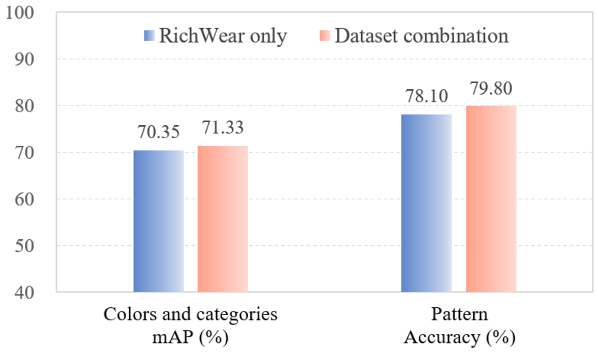


FIGURE 10. Comparison of FARNet model performance when trained on RichWear and on RichWear + Fashion550k.

the very noisy group has the largest improvement, which shows our model's potential to correct the label noise for very noisy classes. In contrast, the very clean group classes have limited room to gain performance improvement because the user-provided labels of these classes are of high quality.

Intuition suggests that a larger training dataset will always be better, and our result in Fig. 10 is consistent with this intuition. Training that uses a combination of RichWear and Fashion550k achieves better performance than that which uses RichWear alone. However, the benefit is relatively small. The accuracy for pattern recognition increases by 1.7 percentage points to 79.8% when Fashion550k is added to the training set. The effect is similar in color and category recognition. In fact, training on RichWear alone is enough to significantly surpass all compared baselines trained on the combined datasets.

While Fashion550k primarily contains street photos from the United States and Europe, it contains street fashion images with clothing attributes in colors and categories that are complementary to RichWear, as mentioned previously. The augmentation of street fashion images increases training samples and provides additional information during the training process, reducing overfitting and leading to better recognition performance. Therefore, training our model by leveraging the manually-corrected images from Fashion550k is consistent with our goal.

VI. STREET FASHION TREND ANALYSIS

In this section, we utilize the fine-grained clothing attributes predicted by our proposed FARNet to analyze street fashion trends from 2017 to 2019. We attempt to find the popularity of clothing colors and patterns in each season. In addition, we perform clustering on the images of RichWear to group visually similar images for street style exploration. Notably, the street fashion trends reported in previous studies focused on American and European styles, while our analysis focuses on Asian fashion.

A. COLOR AND PATTERN TRENDS

We applied our trained model on the full RichWear dataset and collected predicted clothing attributes for street fashion trend analysis. The clothing color trends (normalized between 0 and 1 using min-max normalization; $x - \min(x)/\max(x) - \min(x)$) and the color frequency distribution are shown in Fig. 11. It is not surprising that clothing color trends usually vary with season. Black and white are the most popular overall colors, regardless of season. Bright colors, such as white and pink, appear more frequently in spring and summer than

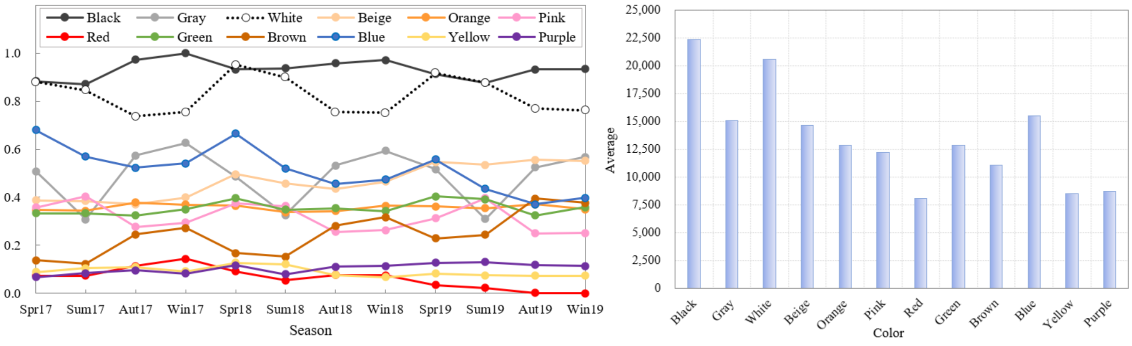


FIGURE 11. Clothing color trends and frequency distribution.

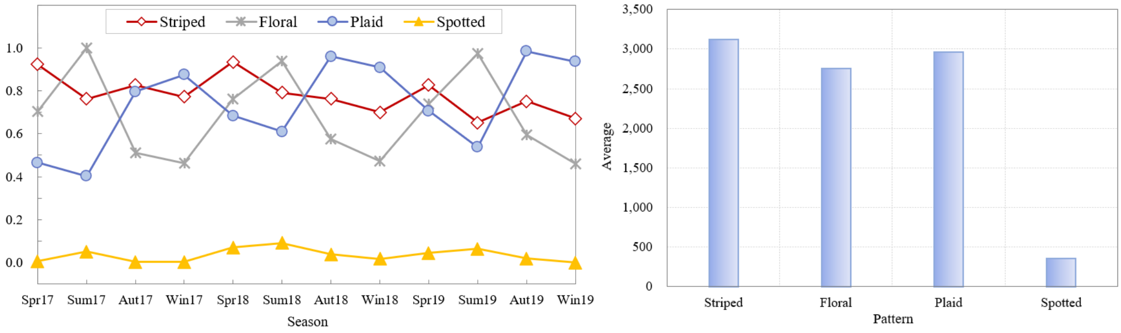


FIGURE 12. Clothing pattern trends and frequency distribution.

in autumn and winter, when darker colors like black, gray, and brown are more popular. While beige becomes popular after 2018, blue gradually goes out of fashion in 2019.

Fig. 12 plots the clothing pattern trends (normalized between 0 and 1 using min-max normalization) and the pattern frequency distribution. We exclude solid here because we focus on trends related to patterned clothing. It is evident that the popularity of individual patterns also changes from season to season. Striped is the most popular pattern in spring, while floral print trends positively in summer. Except for autumn 2017, people wear more plaid clothing in autumn and winter. Compared with other patterns, spotted is less popular, but it also exhibits a seasonal variation.

B. IMAGE CLUSTERING

To discover Asian street styles from 2017 to 2019, we clustered the images of our RichWear dataset, hypothesizing that the visually-correlated images in a cluster may reveal a common street style. The representations used in the clustering analysis are the 2,048-dimensional latent image features extracted from the last layer of the base CNN and normalized by L2-normalization. We considered several common clustering algorithms, including Gaussian mixture model (GMM), K-means, agglomerative hierarchical clustering (AHC), and density-based spatial clustering of applications with noise (DBSCAN). We fine-tuned the number of clusters for each algorithm except for DBSCAN, which does not require this input. To determine the most appropriate clustering algorithm for our analysis, we developed a human evaluation method named ''image intrusion task'' that is inspired by the ''word intrusion task'' proposed by [42].

Specifically, let $\mathbf{x}_i$ denote the 2,048-dimensional feature vector; GMM assumes that the observed $\mathbf{x}_i$ is generated from a mixture of M component Gaussian densities [43]:

$$p(\mathbf{x}_i|w_m, \boldsymbol{\mu}_m, \Sigma_m) = \sum_{m=1}^{M} w_m\, g(\mathbf{x}_i|\boldsymbol{\mu}_m, \Sigma_m), \tag{9}$$

where w_m is the mixture weight with the constraint of $\sum_{m=1}^{M} w_m = 1$, $g(\cdot)$ is the Gaussian density, $\boldsymbol{\mu}_m$ is the mean vector, and Σ_m is the covariance matrix. We use GMMs with full covariance matrices, so each component has its own general covariance matrix.

To select the number of clusters, we used the Bayesian Information Criterion (BIC) to determine M for GMM. The BIC is defined as $-2\ln(\hat{L}) + d \cdot \ln(N)$, where $\hat{L}$ is the maximum value of the likelihood function, d is the number of free parameters to be estimated, and N is the sample size of data. The likelihood of a GMM increases monotonically with the number of components. We chose the number of clusters where the change of the slope of BIC curve is the largest, so there is no much information gain from increasing the number of clusters. K-means and AHC are not associated with likelihood, so we adopted the silhouette coefficient [44] to determine the number of clusters.

In order to choose the most appropriate clustering method for subsequent analysis, we developed the image intrusion task. The image intrusion task assumes that better clustering results are more cohesive in terms of clothing similarity. In the image intrusion task, the participants were presented with six randomly-ordered images, and they were instructed to select an image that is "out of place" in terms of clothing similarity. We asked participants to ignore image backgrounds, human poses, or anything else irrelevant to clothing properties.

Fig. 13 presents an example of the task. Among the six images, the character in Image 3 is wearing pants, while the characters in other images are wearing skirts. Image 3 is clearly out of place and is the intruder to this set of images. We noticed in our preliminary experiments that participants tend to randomly select an intruder if the set of images lack clothing coherence.

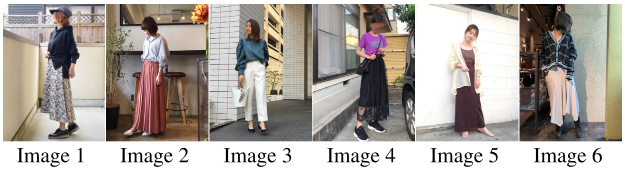


FIGURE 13. An example of the image intrusion task. Image 3 is the intruder to this set of images.

To prepare a set of images to present to the participants, we first compute the centroid of a cluster and randomly select five of K images nearest that centroid ($K = 12$). We then randomly select a different cluster and add one more image I' from this cluster to form a set of six. The image I' is also selected randomly from among K images closest to its centroid. Using this approach, we construct a set of six images for each cluster that is generated by the above-mentioned clustering algorithms.

We obtain the image intrusion score for a clustering algorithm by computing the average proportion of clusters in which participants correctly identified intruders. Intuitively speaking, if a clustering algorithm generates more cohesive clusters, then it should be easier for human participants to correctly identify the intruding images. As a result, the higher image intrusion score indicates better clustering performance. For this task, we recruited 36 participants, all of whom completed several sets of tutorial examples before engaging the task.

Table 5 summarizes the image cluster evaluation of all images in the RichWear dataset. The second column lists the number of clusters for each clustering algorithm. The number of clusters derived from DBSCAN is 45, which is the highest among all clustering algorithms. GMM, K-means, and AHC have 31, 20, and 7 clusters, respectively. The image intrusion task shows that GMM has the most cohesive clustering results with a score of 0.90. The score means that, on average, the participants can correctly identify intruding images in 90% of clusters. DBSCAN, on the other hand, is the worst in terms of cluster coherence. The score is only 0.16, which means that the participants can only correctly identify intruding images in 16% of clusters. K-means and AHC have a score of 0.60 and 0.41, respectively. Both are behind GMM by a large gap.

Based on the results of the human evaluation method, we adopted GMM to explore Asian street fashions by gathering visually-correlated images in RichWear. Based on the BIC criterion, we set $M = 16$ for clustering by year to discover fine-grained fashion trends. Moreover, we set $M = 31$ for clustering on the full dataset, as just mentioned, to perceive the temporal dynamics of identified street styles.

1) FASHION TRENDS

We clustered the street fashion images in each subset of RichWear by year in order to discover fashion trends. Except for a few noisy clusters that are disturbed by human posing or patterned backgrounds, most clusters contain the images with one clear fashion. Table 6 shows the street fashion trends in Japan and other Asian areas from 2017 to 2019. We discover that some common street fashions last for three years, such as long outerwear and black clothing. In contrast, some street fashions only appear in a single year, such as women's oversized clothes in 2017, shirt and shirt jacket in 2018, and maxi skirt and maxi dress in 2019. We provide more clustering results in Section A of the supplemental material.

In addition, user-created hashtags on the fashion website represent common styles among users. To find the relevant styles for a street fashion, we use the following approach: (1) We count the five most common hashtags of the images within a cluster, (2) we count hashtags with the same meaning only once, (3) we omit the hashtags of clothing categories, like #Dress and #Sneakers, and (4) we translate a few Japanese hashtags into English and give the original hashtags in parenthesis. Finally, we show the five most common hashtags for each street fashion in Table 6. For example, the most relevant style hashtag associated with long outerwear is #AdultCasual. We observe that the #Simple and #AdultCasual hashtags are generally the most popular for street fashions.

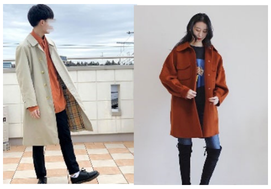
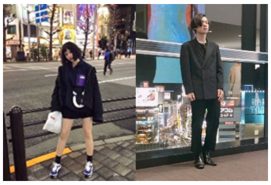
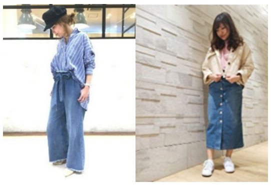
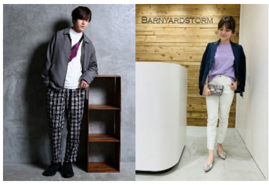
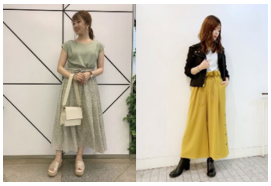
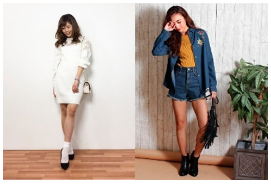
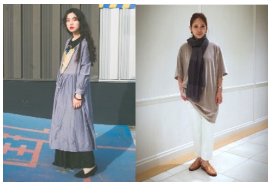
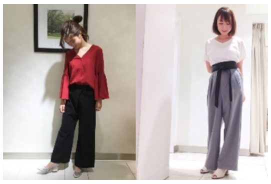
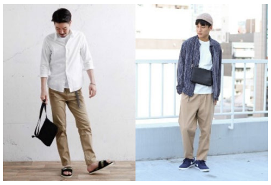
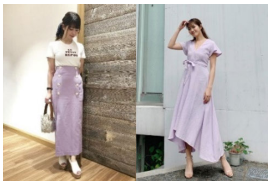
2) STYLE DYNAMICS

We attempt to identify some street styles from our dataset and to find the temporal dynamics of these styles. For street style dynamics discovery, we perform clustering on the full RichWear dataset. After clustering, we can identify clusters that reveal dominant clothing styles. There are also several noisy clusters that contain no common style, which are disturbed by human posing or patterned backgrounds. A few popular clothing styles appear in more than one cluster, such as jeans and other denim clothes. We choose only one cluster among those with the same style to analyze its style dynamics. In addition, although some clusters contain images with a dominant clothing style, such as pants coordinates, their style dynamics do not show significant seasonal fluctuations. We thus exclude the dynamics of these regular clothing styles and finally choose seven clusters to present their temporal

TABLE 5. Objective and Human Evaluation of Image Clusters.

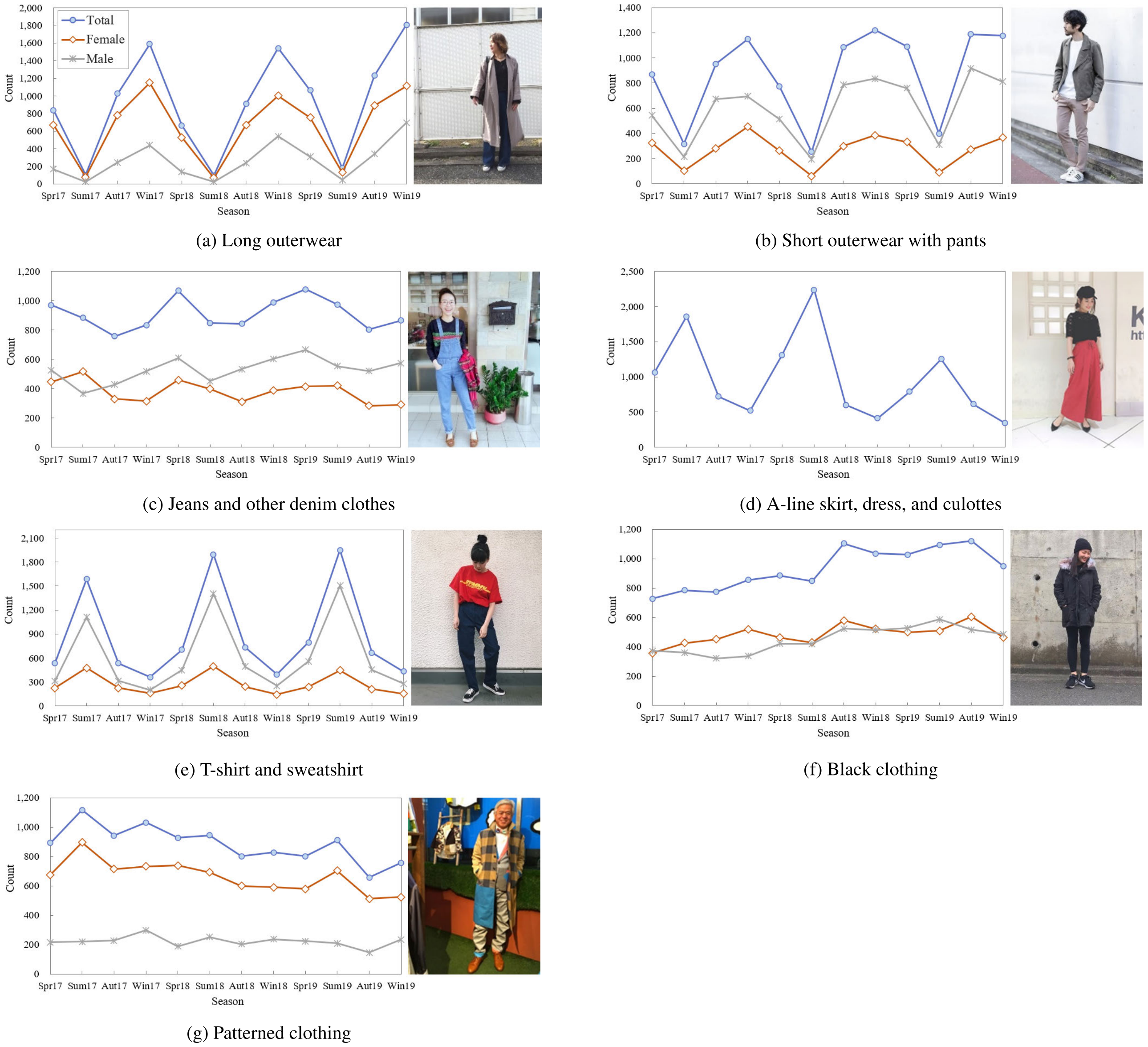
Clustering Algorithm	Number of Clusters	Image Intrusion Score (Human Evaluation)
K-means	20	0.60
GMM	31	0.90
AHC	7	0.41
DBSCAN	45	0.16

TABLE 6. The street fashion trends and their relevant style hashtags from 2017 to 2019.

• Long outerwear (2017–2019)  #AdultCasual (大人カジュアル) #Simple (シンプル) #WinterOutfit (冬コーデ) #OOTD (Outfit of the Day) #ClassyCasual (きれいめカジュアル)	• Black clothing (2017–2019)  #Simple (シンプル) #AdultCasual (大人カジュアル) #Casual (カジュアル) #FashionCommunity (おしゃれさんと繋がりたい) #HolidayStyle (休日スタイル)
• Jeans and other denim clothes (2017–2019)  #Simple (シンプル) #Casual (カジュアル) #AdultCasual (大人カジュアル) #HolidayStyle (休日スタイル) #CheapFashion (プチプラ)	• Short outerwear with pants (2017–2019)  #Simple (シンプル) #AdultCasual (大人カジュアル) #FashionCommunity (おしゃれさんと繋がりたい) #WinterOutfit (冬コーデ) #SpringOutfit (春コーデ)
• A-line skirt, dress, and culottes (2017–2019)  #AdultCasual (大人カジュアル) #Casual (カジュアル) #Simple (シンプル) #WomenStyle (オトナ女子) #OOTD (Outfit of the Day)	• Mini skirt and short dress/pants (2017)  #Simple (シンプル) #AdultCasual (大人カジュアル) #Casual (カジュアル) #SummerOutfit (夏コーデ) #CheapFashion (プチプラ)
• Women's oversized clothes (2017)  #AdultCasual (大人カジュアル) #Casual (カジュアル) #Simple (シンプル) #AutumnOutfit (秋のコーデ) #SpringOutfit (春コーデ)	• Plain clothing (2017)  #Simple (シンプル) #AdultCasual (大人カジュアル) #AutumnOutfit (秋のコーデ) #Neat (ニット) #ClassyCasual (きれいめカジュアル)
• Shirt and shirt jacket (2018)  #Simple (シンプル) #AdultCasual (大人カジュアル) #Casual (カジュアル) #FashionCommunity (おしゃれさんと繋がりたい) #AutumnOutfit (秋のコーデ)	• Maxi skirt and maxi dress (2019)  #AdultCasual (大人カジュアル) #SpringOutfit (春コーデ) #Casual (カジュアル) #Petite (低身長) #Simple (シンプル)

dynamics. We provide image examples of the clustering result in Section B of the supplemental material. The users' gender information is also utilized to identify the differences between men's and women's street fashion trends.

Fig. 14 shows the street style dynamics results. Each style has seasonal fluctuations in frequency that reveal popularity and unpopularity over time as well as several interesting observations: (1) Long outerwear is more popular in winter than in other seasons, but short outerwear with pants is popular not only in winter but also in spring and autumn. Additionally, long outerwear is more popular among women, while short outerwear with pants is more popular among



(a) Long outerwear

(b) Short outerwear with pants

(c) Jeans and other denim clothes

(d) A-line skirt, dress, and culottes

(e) T-shirt and sweatshirt

(f) Black clothing

(g) Patterned clothing

FIGURE 14. Street style dynamics.

men. (2) People wear more jeans and other denim clothes in spring. Men wore more jeans and other denim clothes in 2019, while women wore less after 2018. (3) A-line skirt, dress, and culottes are popular in summer, especially in the summer of 2018.[5] (4) T-shirt and sweatshirt have spikes in frequency during the summer, and seasonal variation is more obvious for men than for women. (5) Black clothing has become increasingly popular, while patterned clothing shows a declining trend. Men wear black clothing as often as women; however, they wear patterned clothing much less than women.

Compared to previous street fashion studies [4], [8], [9], we have discovered unique street fashion trends in Japan and other Asian areas. We have also integrated the user-created hashtags in our dataset to further understand relevant styles for street fashions. Moreover, we have successfully identified street style dynamics and differences between men's and women's fashion styles that have not been explored in previous studies.

VII. CONCLUSION

This study aimed to explore Asian street fashion by creating a novel fashion recognition architecture and a large-scale image dataset with user-provided noisy labels, and it contributes to the existing literature in three areas. First, this

[5] Since A-line skirt, dress, and culottes are women's clothing, we do not discuss this trend by gender.

study developed a new street fashion dataset named RichWear, which contains 322,198 street fashion images with upload date, users' gender and country, clothing brands, and user-created hashtags. In addition to user-provided noisy labels, we also created a 4,368-image subset with expert-verified labels for three types of clothing attributes. In particular, RichWear focuses on street styles in Japan and other Asian areas, providing a good data source for Asian fashion understanding.

For fashion recognition, we have proposed a multi-task neural network, FARNet, which can leverage noisy labels and simultaneously recognize multiple clothing attributes. This network facilitates our street fashion exploration in the large-scale dataset collected from a social media site. The Noise Correction Network in FARNet is based on an existing network, but it more effectively corrects for noisy labels. It achieves better performance (71.33%) than the compared baselines (55.46%–59.79%). Moreover, our empirical results show that MTL, when compared to STL, can noticeably improve generalization performance of attribute recognition.

Finally, by using both supervised and unsupervised learning methods, we have documented interesting street fashion trends in Asia from 2017 to 2019. We have also observed significant seasonal dynamics for men's and women's street styles that have not been explored in previous studies. In future work, we plan to incorporate product and brand information to further refine the fashion trend analysis. We are interested in the mercurial popularity of products and brands, a deeper understanding that may help us predict the rise and fall of a particular product, brand, or style.

REFERENCES

[1] N. Liu, S. Ren, T.-M. Choi, C.-L. Hui, and S.-F. Ng, "Sales forecasting for fashion retailing service industry: A review," *Math. Problems Eng.*, vol. 2013, Nov. 2013, Art. no. 738675.

[2] Y. Guan, Q. Wei, and G. Chen, "Deep learning based personalized recommendation with multi-view information integration," *Decis. Support Syst.*, vol. 118, pp. 58–69, Mar. 2019.

[3] M. F. Hashmi, B. K. K. Ashish, A. G. Keskar, N. D. Bokde, and Z. W. Geem, "FashionFit: Analysis of mapping 3D pose and neural body fit for custom virtual try-on," *IEEE Access*, vol. 8, pp. 91603–91615, 2020.

[4] X. Gu, Y. Wong, P. Peng, L. Shou, G. Chen, and M. S. Kankanhalli, "Understanding fashion trends from street photos via neighbor-constrained embedding learning," in *Proc. 25th ACM Int. Conf. Multimedia*, Mountain View, CA, USA, Oct. 2017, pp. 190–198.

[5] N. Inoue, E. Simo-Serra, T. Yamasaki, and H. Ishikawa, "Multi-label fashion image classification with minimal human supervision," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 2261–2267.

[6] L. Lo, C.-L. Liu, R.-A. Lin, B. Wu, H.-H. Shuai, and W.-H. Cheng, "Dressing for attention: Outfit based fashion popularity prediction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 3222–3226.

[7] Y. Ma, J. Jia, S. Zhou, J. Fu, Y. Liu, and Z. Tong, "Towards better understanding the clothing fashion styles: A multimodal deep learning approach," in *Proc. Assoc. Advancement Artif. Intell. (AAAI)*, San Francisco, CA, USA, 2017, pp. 38–44.

[8] K. Matzen, K. Bala, and N. Snavely, "StreetStyle: Exploring worldwide clothing styles from millions of photos," 2017, *arXiv:1706.01869*. [Online]. Available: http://arxiv.org/abs/1706.01869

[9] U. Mall, K. Matzen, B. Hariharan, N. Snavely, and K. Bala, "GeoStyle: Discovering fashion trends and events," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 411–420.

[10] M. Takagi, E. Simo-Serra, S. Iizuka, and H. Ishikawa, "What makes a style: Experimental analysis of fashion prediction," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 2247–2253.

[11] W.-L. Hsiao and K. Grauman, "Learning the latent 'Look': Unsupervised discovery of a style-coherent embedding from fashion images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 4203–4212.

[12] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun, "Neuroaesthetics in fashion: Modeling the perception of fashionability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 869–877.

[13] K. Yamaguchi, M. H. Kiapour, and T. L. Berg, "Paper doll parsing: Retrieving similar styles to parse clothing items," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 3519–3526.

[14] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Parsing clothing in fashion photographs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3570–3577.

[15] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu, "ModaNet: A large-scale street fashion dataset with polygon annotations," in *Proc. 26th ACM Int. Conf. Multimedia*, Seoul, South Korea, 2018, pp. 1670–1678.

[16] W. Yang, P. Luo, and L. Lin, "Clothing co-parsing by joint image segmentation and labeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3182–3189.

[17] J. Xiang, T. Dong, R. Pan, and W. Gao, "Clothing attribute recognition based on RCNN framework using L-Softmax loss," *IEEE Access*, vol. 8, pp. 48299–48313, 2020.

[18] X. Zhang, J. Jia, K. Gao, Y. Zhang, D. Zhang, J. Li, and Q. Tian, "Trip outfits advisor: Location-oriented clothing recommendation," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2533–2544, Nov. 2017.

[19] B. Zhao, J. Feng, X. Wu, and S. Yan, "Memory-augmented attribute manipulation networks for interactive fashion search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1520–1528.

[20] I. Tautkute, T. Trzcinski, A. P. Skorupa, K. Lukasz, and K. Marasek, "DeepStyle: Multimodal search engine for fashion and interior design," *IEEE Access*, vol. 7, pp. 84613–84628, 2019.

[21] P. Li, Y. Li, X. Jiang, and X. Zhen, "Two-stream multi-task network for fashion recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 3038–3042.

[22] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1096–1104.

[23] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 839–847.

[24] J. Han, P. Luo, and X. Wang, "Deep self-learning from noisy labels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 5138–5147.

[25] MessyCloset. (2020). *What is Street Style*. [Online]. Available: https://www.messycloset.com/fashion/street-style/864/

[26] E. Simo-Serra and H. Ishikawa, "Fashion style in 128 floats: Joint ranking and classification using weak data for feature extraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 298–307.

[27] K. Yamaguchi, T. L. Berg, and L. E. Ortiz, "Chic or social: Visual popularity analysis in online fashion networks," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, 2014, pp. 773–776.

[28] S. Vittayakorn, K. Yamaguchi, A. C. Berg, and T. L. Berg, "Runway to realway: Visual analysis of fashion," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Waikoloa, HI, USA, Jan. 2015, pp. 951–958.

[29] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg, "Hipster wars: Discovering elements of fashion styles," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland, 2014, pp. 472–488.

[30] S. H. Lee, C. H. Moon, and T. L. N. Tu, *Fashion and Beauty in the Time of Asia*. New York, NY, USA: NYU Press, 2019, pp. 7–8.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 37. Lille, France, 2015, pp. 448–456.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[35] D. Pedamonti, "Comparison of non-linear activation functions for deep neural networks on MNIST classification task," 2018, *arXiv:1804.02763*. [Online]. Available: http://arxiv.org/abs/1804.02763

[36] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 2682–2690.

[37] I. Misra, C. L. Zitnick, M. Mitchell, and R. Girshick, "Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2930–2939.

[38] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[39] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 21–37.

[41] Z. Meng, X. Fan, X. Chen, M. Chen, and Y. Tong, "Detecting small signs from large images," in *Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI)*, San Diego, CA, USA, Aug. 2017, pp. 217–224.

[42] J. Chang, S. Gerrish, C. Wang, J. Boyd-Graber, and D. Blei, "Reading tea leaves: How humans interpret topic models," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, 2009, pp. 288–296.

[43] D. Reynolds, "Gaussian mixture models," in *Encyclopedia Biometrics*. Boston, MA, USA: Springer, 2015, pp. 827–832.

[44] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

FU-HSIEN HUANG received the M.S. degree in statistics from National Taiwan University, Taipei, Taiwan, in 2020. Her research interests include machine learning, computer vision, and statistical analysis.

HSIN-MIN LU (Member, IEEE) received the Ph.D. degree from the University of Arizona. He is currently an Associate Professor with the Department of Information Management, National Taiwan University. His research interests include data mining, machine learning, medical informatics, and business analytics.

YAO-WEN HSU received the Ph.D. degree. He is currently an Associate Professor with the Department of International Business and the Master Program in Statistics with National Taiwan University. He has published research articles in international journals, including *Decision Support Systems*, the *Journal of Systems and Software*, *Environment International*, *Annals of Operations Research*, and *Small Business Economics*. His research interests include big data analysis, risk management, and revenue management in the hospitality industry.