Photo Filter Classification and Filter Recommendation without Much Manual Labeling

Wei-Ta Chu
National Cheng Kung University
Tainan, Taiwan
wtchu@gs.ncku.edu.tw

Yu-Tzu Fan
National Chung Cheng University
Chiayi, Taiwan
p3212321@gmail.com

***Abstract*—Because how users employ filters to photos may reveal user's preference or mental state, a photo filter classification method is potentially demanded to enable future large-scale analysis. We adopt the transfer learning technique to transform deep models pre-trained for object classification into models suitable for photo filter classification. Based on accurate classification results, we build a filter recommendation approach without much manual labeling. It can be easily extended when more training data are available. A series of experimental studies are conducted to demonstrate effectiveness of filter classification with transfer learning. We also demonstrate the proposed filter recommendation achieves encouraging performance.**

***Index Terms*—Photo filter classification, photo filter recommendation, matrix factorization**

I. Introduction

As the flourishing growth of social media platforms, tremendous amounts of photos like selfies and pictures taken in a journey are uploaded and shared online. To stylize photos or make them more attractive, various filters are employed by users. The study in [1] shows filtered photos would receive more views and comments. Because of the impact of photo filtering, many social platforms and mobile apps have provided such functionalities.

Many photos shared on social media platforms are filtered. Due to the demands of employing filtering to photos, photo filter recommendation has been proposed based on aesthetic learning [2]. Given a pair of photos that have the same visual content but are with different filters, aesthetic features of these photos are extracted by convolutional neural networks. Associated with the category information, the filter yielding a higher aesthetic score is recommended to users. More recently, Wu et al. [3] modeled semantic mappings between color themes and commonly-used keywords. Photo filters are then organized by the derived semantic information, so that users can more easily find the filters that matches with the desired characteristics.

In this work, we target at two tasks: photo filter classification and photo filter recommendation. Recently, Reece and Danforth [4] showed that depressive signals can be detected via analyzing the filtered photos posted on Instagram. The adopted filters not only show user's preference but also reveal user's mental state. In the era of big data and social media, knowing mental state of users becomes extremely valuable. Motivated by this interesting study, we think a photo filter classification method is required to enable future large-scale analysis.

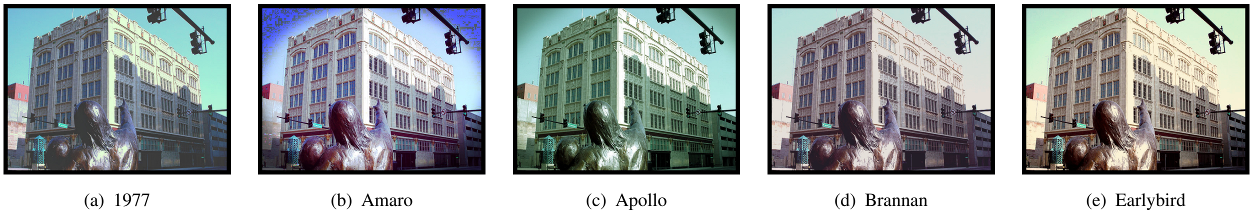
Despite the request of photo filter classification, accurately predicting the filter applied to a given photo is not a trivial task. First, by varying processing parameters, such as contrast, saturation, and exposure, a large number of photo filters have been created. There are over 40 filters natively provided by Instagram, for example. Second, the differences between different filters are sometimes subtle and cannot be easily recognized. Figure 1 shows five examples of the same photos applied with five different filters. The visual effects of Brannan and Earlybird are similar, for example. Third, most current classification models are designed for object or scene classification. As shown in Figure 1, the five examples show the same content but with different visual effects. Therefore, the main challenge would be how to transform the current object-oriented or scene-oriented classification models into filter-oriented classification models.

In this paper, we propose to achieve model transformation by the transfer learning technique. Given a model pre-trained based on a large-scale data collection, we will take the pre-trained model parameters as the initial settings, and fine-tune the model based on the collected photo filter dataset. Performance variations yielded by different learning schemes will be investigated.

For photo filter recommendation, we propose an easy-to-extend approach on the basis of photo filter classification. Selecting a photo filter of interest is quite subjective. To build a deep learning method like [2], a large collection of user preference is needed for training. However, collecting user preference, even on Amazon Mechanical Turk, is costly and time-consuming. We propose to collect a large number of photos, and determine each photo's filter by the developed (very accurate) filter classifier. Based on this image collection and associated filter types, we adopt matrix factorization techniques conventionally used in text-based recommender systems to achieve filter recommendation. Notice that this approach does not need to collect user preference. Through the association between photo's visual content and adopted filters, we indirectly make users providing the collected photos collaborate. No costly manual labeling is needed.

The rest of this paper is organized as follows. Section II

978-1-7281-1817-8/19/$31.00 ©2019 IEEE



(a) 1977 (b) Amaro (c) Apollo (d) Brannan (e) Earlybird

Fig. 1. Five different filters (1977, Amaro, Apollo, Brannan, and Earlybird) are applied to the same photo (better viewed in color).

introduces photo filters and the collected evaluation dataset. Section III describes details of the transfer learning scheme adopted to build the filter classifier. Based on classification results, Section IV presents details of filter recommendation. In Section V, we demonstrate performance variations of different settings and make discussion. Section VI makes the concluding remarks.

II. Photo Filters and Dataset

To conduct filter classification, we take the FACD (Filter Aesthetic Comparison Dataset) [2] as the basis and make an extension. Images in FACD were sampled from the eight most popular categories of the AVA dataset [5]. These categories are *animal*, *flora*, *landscape*, *architecture*, *food and drink*, *portrait*, *cityscape*, and *still life*. From each category, 160 unfiltered photos were randomly sampled. Twenty-two filters[1] provided by both GNU Image Manipulation Program (GIMP) toolkit[2] and Instagram[3] were then applied to each selected photo, yielding 28,160 filtered photos in total. The photos' resolution is 227×227.

To make a larger evaluation dataset for filter classification, we further sample 320 unfiltered photos from each of the eight categories of the AVA dataset, and apply the 22 filters to these photos to get 56,320 filtered photos in total. Combining these manually-collected data with FACD, we totally have 84,480 filtered photos for the study of filter classification.

For filter recommendation, we will adopt the matrix factorization techniques, and need to construct matrices based on visual features and recognized filter types of collected photos. We collect 16,256 photos in total from Instagram users who are familiar with using photo filters. These users include celebrities, designers, and so on. All photos are resized to 224×224. Filter of each photo will be determined by the developed filter classifier, and this information will be utilized to build the filter recommender system.

III. Photo Filter Classification

Comparing with large-scale datasets like ImageNet [6], AVA [5], and MSCOCO [7], our collected dataset is much smaller. To well take advantage of the power of deep models pre-trained on a large-scale dataset, we would like to adopt transfer learning techniques and transform a pre-trained model into updated one that is more suitable to photo filter classification.

In this work, we investigate transferring three popular convolutional neural networks, i.e., AlexNet [8], VGG-16 [9], and ResNet-50 [10], pre-trained on the ImageNet dataset and were originally developed for object recognition. How these models can be transferred to do photo filter classification will be studied experimentally.

AlextNet [8] consists of five convolutional layers followed by three fully-connected layers. The output of the last fully-connected layer is fed to a 1000-way softmax to produce the probabilities of a given image being the 1000 classes (defined by the ImageNet dataset). VGG-16 [9] consists of 13 convolutional layers followed by three fully-connected layers. The 13 convolutional layers can be divided into five groups by four max pooling layers. The output of the last fully-connected layer is similar to that of AlexNet. Deeper networks are more difficult to train. He et al. [10] demonstrated that deeper networks (more layers) don't necessarily yield better performance. They thus proposed stacking residual blocks to constitute a deeper network and keep decreasing training/testing errors. In this work, we adopt the ResNet-50 model that contains 50 layers in total.

To fine-tune the pre-trained models, from each filter type we randomly select 80% of filtered photos as the training data, and the remaining 20% are taken as the testing data. The five-fold cross validation scheme is adopted. Therefore, we will run the training/testing process five times and report the average test accuracy. The last fully-connected layers of these pre-trained models are modified to a 22-way softmax, in accordance with the number of photo filters.

Table I shows the empirical fine-tuning parameters we adopted to achieve transfer learning. To fine-tune AlexNet, the size of mini-batch is 128, while the size is 64 for VGG-16 and ResNet-50. In fine-tuning ResNet-50, we found a much smaller learning rate should be set to get better results. The loss function for fine-tuning these models is cross entropy.

Previous studies related to filter classification are not many. Bianco [11] evaluated three famous convolutional neural networks for filter classification, and demonstrated promising performance can be obtained. However, only place images were included in the evaluation, and generality of these networks were not shown. On the other hand, Chen et al. [12] worked on filter-invariant image classification. They developed a Siamese network based on the designed loss to solve the problem of filter bias.

[1] Names of 22 filters: 1977, Amaro, Apollp, Brannan, Earlybird, Gotham, Hefe, Hudson, Inkwell, Lofi, LordKevin, Mayfair, Nashville, Poprocket, Rise, Sierra, Sutro, Toaster, Valencia, Walden, Willow, and XProII.

[2] https://www.gimp.org

[3] https://www.instagram.com

TABLE I
PARAMETERS OF FINE-TUNING PRE-TRAINED MODELS.

Models	Learning rate	Iteration	Loss
AlexNet	0.001	63240	cross entropy
VGG-16	0.001	63300	cross entropy
ResNet-50	0.000001	63300	cross entropy

IV. Photo Filter Recommendation

To build a filter recommender system, one way is developing a learning method that takes photos as input and output predicted user preference. This approach needs a large amount of photos with associated user preference. For example, Sun et al. [2] put image pairs on Amazon Mechanical Turk and asked online workers to do pairwise comparison. The collected user annotations need more processes to ensure the quality and make consensus. Therefore, this approach is costly from the perspective of ground truth data collection.

In this work, we decide to adopt matrix factorization techniques to achieve filter recommendation. Based on the filter classifier mentioned above, we can recognize each photo's filter type. In addition, we can extract visual information, such as object and scene, from each photo. Based on a set of photos, each of which is with a filter type and a visual descriptor, we can construct a $N \times M$ matrix $\boldsymbol{P}$ showing co-occurrence of filter types and visual descriptors, where N is the number of different filter types and M is the dimensionality of visual vectors. Initially all row vectors $\boldsymbol{p}_j$, $j = 1, 2, ..., N$, are zero vectors. For a photo with filter type i and with the visual descriptor $\boldsymbol{v} = (v_1, v_2, ..., v_M)$, we add $\boldsymbol{v}$ to the ith row vector $\boldsymbol{p}_i$ of $\boldsymbol{P}$, i.e., $\boldsymbol{p}_i = \boldsymbol{p}_i + \boldsymbol{v}^T$. By accumulating information from all training photos, we can build the matrix $\boldsymbol{P}$. We can keep collecting photos from the internet as much as possible, and continuously enrich the matrix $\boldsymbol{P}$ without manually labeling user's preference. This is the most important advantage of the proposed approach.

In the following, we first describe the adopted matrix factorization technique, and then describe three visual descriptors to construct co-occurrence matrices.

A. Matrix Factorization

Matrix factorization (MF) [13] is a widely adopted method in recommender systems. Given a matrix showing how users select/collect items, the MF approach discovers latent relationships between users and items, and then predicts how likely a user would select an item that has never seen before. In our work, the matrix $\boldsymbol{P}$ we provide shows how strongly or how frequently photos with a specific filter convey predefined visual descriptors. For example, how frequently photos with *Amaro* filtering effect present objects like *person*, *dog*, *chair*, and so on. Given the matrix $\boldsymbol{P}_{N\times M}$, where N is the number of filter types and M is the number of visual descriptors, the prediction task is to find two matrices $\boldsymbol{X}_{N\times K}$ and $\boldsymbol{Y}_{M\times K}$ such that their product approximates $\boldsymbol{P}$:

$$\boldsymbol{P} \approx \boldsymbol{X} \times \boldsymbol{Y}^T = \hat{\boldsymbol{P}}. \tag{1}$$

This process maps filter type and visual descriptors to a K-dimensional latent factor space. The elements of the ith row vector $\boldsymbol{x}_i$ of $\boldsymbol{X}$ represent the strength of the association between the ith filter type and the latent factors. The elements of the jth row vector $\boldsymbol{y}_j$ of $\boldsymbol{Y}$ represent the strength of the association between the jth visual descriptor and the latent factors. In this work, we conduct nonnegative matrix factorization [14] by the function implemented in MATLAB.

Given a photo, we extract its M-dimensional visual descriptors as $\boldsymbol{v}$, and then embed it into the latent factor space as $\boldsymbol{y}_v = \boldsymbol{Y}^T\boldsymbol{v}$. How likely filters are recommended to this photo is calculated as:

$$\hat{\boldsymbol{p}} = \boldsymbol{X}\boldsymbol{y}_v. \tag{2}$$

Based on $\hat{\boldsymbol{p}} = (\hat{p}_1, \hat{p}_2, ..., \hat{p}_N)$, we can determine the final recommended filter type i^* by finding the one with the maximum predicted value, i.e., $i^* = \arg\max_i \hat{p}_i$.

B. Visual Information

We describe visual information of photos from both global and local perspectives. Four features are extracted, including visual features derived from an autoencoder (global), place information (global), aesthetics information (global), and object information (local). We describe details of them in the following.

Auto features. Given an image $\boldsymbol{x}$, an autoencoder first encodes it into a (usually) lower-dimensional vector $\boldsymbol{x}' = f(\boldsymbol{x})$, which is then decoded to a vector $\boldsymbol{x}'' = g(\boldsymbol{x}')$ of the same dimension as the input. The goal of this autoencoder is to make the decoded vector $\boldsymbol{x}''$ as similar to the input $\boldsymbol{x}'$ as possible. In this work, we build the encoding function f by a series of convolutional layers followed by one fully-connected layer. The first and the second convolutional layers output 16 and 32 feature maps, respectively, and the convolution kernel for both layers is 5×5 with stride 2. Outputs of the second convolutional layer are flattened and concatenated as a high-dimensional vector, and this vector is reduced to a 1024-dimensional (1024-dim in short) embedding vector by a fully-connected layer. The decoding function g, on the other hand, is built by a fully-connected layer followed by two deconvolutional layers. The loss function to train the autoencoder is mean square error, and this autoencoder is trained based on the collected 16,256 photos mentioned in Sec. II. Overall, the vector output by f is viewed as the features of $\boldsymbol{x}$. Motivated by the bag of word model, we collect features extracted from the training data, and cluster them by the K-means algorithm ($K = 40$ in this work). Each feature vector is then categorized into one of the forty clusters, and is encoded as a 40-dim one-hot binary vector, which is the final visual descriptor we call Auto features.

The matrix factorization technique mentioned in Sec. IV-A is just the baseline. According to [15], the effect of factorization can be boosted if the input matrix is enhanced first. Motivated by this idea, we apply principal component analysis (PCA) to reduce the input matrix $\boldsymbol{P}_{N\times 40}$ into $\boldsymbol{P}_{N\times 20}$. After this initialization, based on the matrix factorization technique, we finally can recommend filters by the estimated values, denoted by $\hat{\boldsymbol{p}}^{(1)} = (\hat{p}_1^{(1)}, \hat{p}_2^{(1)}, ..., \hat{p}_N^{(1)})$.

Place information. Some photo filters may be more suitable to be applied to specific places. Therefore, we adopt the ResNet-50 model trained on the Place365 dataset [16] to estimate how likely the given photo was taken in some specific places. Particularly, given an image $\boldsymbol{x}$, the pretrained ResNet-50 model outputs the probability vector showing the strength that $\boldsymbol{x}$ is in the 365 places. Examples of these places include *cafeteria*, *street*, *garden*, *rainforest*, and so on. Results of place recognization are 365-dim probability vectors. Considering that the number of place is large and some places are correlated (e.g., cottage garden, herb garden, and botanical garden), we cluster these vectors by the K-means algorithm, and encode them into a 40-dim one-hot vector. The input matrix $\boldsymbol{P}_{N\times 40}$ is then constructed and is reduced into $\boldsymbol{P}_{N\times 20}$ by PCA. Based on the matrix factorization technique, we can recommend filters by the estimated values, denoted by $\hat{\boldsymbol{p}}^{(2)} = (\hat{p}_1^{(2)}, \hat{p}_2^{(2)}, ..., \hat{p}_N^{(2)})$.

Aesthetics Information. The work in [2] showed that filter selection is related to aesthetics information. They adopted the RAPID net proposed in [17] to extract aesthetic features. We also want to extract aesthetic features here, and adopt a more recent model called NIMA proposed in [18] to do so. Particularly, given an image $\boldsymbol{x}$, the NIMA model on the basis of the efficient MobileNet structure outputs 1024-dimensional feature vectors. Similar to the process mentioned above, we cluster these vectors by the K-means algorithm, and encode them into a 40-dim one-hot vector. The input matrix $\boldsymbol{P}_{N\times 40}$ is then constructed and is reduced into $\boldsymbol{P}_{N\times 20}$ by PCA. Based on the matrix factorization technique, we can recommend filters by the estimated values, denoted by $\hat{\boldsymbol{p}}^{(3)} = (\hat{p}_1^{(3)}, \hat{p}_2^{(3)}, ..., \hat{p}_N^{(3)})$.

Object information. The aforementioned information is globally embedded in photos. Here we adopt the YOLOv3 [19] object detector to extract local information showing on objects. Given an image $\boldsymbol{x}$, the YOLOv3 model outputs a 80-dim probability vector showing the probabilities of 80 predefined objects present in this image. Similarly, we cluster probability vectors obtained from the training data by the K-means algorithm, and each vector is then categorized into one of the forty clusters. A 40-dim one-hot binary vector is also obtained as the object descriptor. The input matrix $\boldsymbol{P}_{N\times 40}$ is then constructed and is reduced into $\boldsymbol{P}_{N\times 20}$ by PCA. Based on the matrix factorization technique, we can recommend filters by the estimated values, denoted by $\hat{\boldsymbol{p}}^{(4)} = (\hat{p}_1^{(4)}, \hat{p}_2^{(4)}, ..., \hat{p}_N^{(4)})$.

We estimate the likelihood of each filter type recommended to the input photo, based on the Auto features, place information, and object information, respectively. The predicted vectors $\hat{\boldsymbol{p}}^{(1)}$, $\hat{\boldsymbol{p}}^{(2)}$, $\hat{\boldsymbol{p}}^{(3)}$, and $\hat{\boldsymbol{p}}^{(4)}$ are fused by linear combination, i.e.,

$$\hat{\boldsymbol{p}} = \lambda_1\hat{\boldsymbol{p}}^{(1)} + \lambda_2\hat{\boldsymbol{p}}^{(2)} + \lambda_3\hat{\boldsymbol{p}}^{(3)} + \lambda_4\hat{\boldsymbol{p}}^{(4)}, \tag{3}$$

where λ_1, λ_2, λ_3, and λ_4 are weightings empirically set as 0.3, 0.4, 0.1, and 0.2, respectively. In the evaluation, we recommend top 1, top 3, and top 5 filter types according to values of $\hat{\boldsymbol{p}}$.

TABLE II
AVERAGE CLASSIFICATION ACCURACY OF DIFFERENT MODELS.

Models	Classification Accuracy
AlexNet (train from scratch)	0.0917
AlexNet (w/o fine-tuning)	0.2778
VGG-16 (w/o fine-tuning)	0.1924
ResNet-50 (w/o fine-tuning)	0.5999
AlexNet (w. fine-tuning)	0.9221
VGG-16 (w. fine-tuning)	0.9565
ResNet-50 (w. fine-tuning)	0.9488

V. EVALUATION

A. Filter Classification

We first try to develop AlexNet's network structure mentioned in [8] and train the model from scratch based on the collected dataset. The first row of Table II shows that this approach works terribly bad. This may be due to the small volume of our dataset (only 84,480 filtered photos). We then evaluate if the models pre-trained on ImageNet are suitable to filter classification. As can be seen, AlexNet and VGG-16 do not work well, while the ResNet-50 achieves much better classification accuracy (around 0.60 accuracy). This may be because ResNet-50 is a much deeper model and is more generic to describe visual content. The design of skip connection in ResNet-50 makes it less overfitting.

After fine-tuning the three models based on the collected dataset and with the parameters mentioned in Table I, we obtain average classification accuracy shown in the third part of Table II. As can be seen, performance of all three models is largely boosted after fine-tuning. This shows the importance of transfer learning. Overall, the VGG-16 model with appropriate fine-tuning achieves the best performance. The average accuracy to classify 22 photo filters is around 95.65%.

As fine-tuning plays an important role in boosting classification accuracy, we would like to investigate if the volume of data for fine-tuning influence performance. Table III shows performance variations yielded by the VGG-16 models fine-tuned with the FACD only and with the entire evaluation dataset. FACD contains 28,160 filtered photos, and the entire evaluation dataset contains 80,520 filtered photos. The values in Table III show that, with more data for fine-tuning, better performance can be achieved. This matches with our expectation.

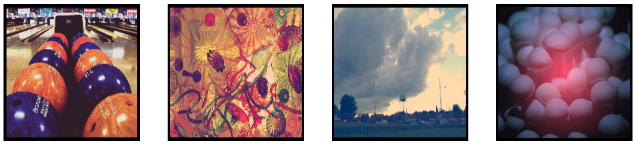
As mentioned previously, the collected photos are from the eight most popular categories of the AVA dataset. We are wondering if photos with different semantics have different filter classification performance. To see this, we collect the statistics of classification accuracies for photos in different categories, based on the VGG-16 model, and show details in Table IV. As can be seen, performance for different semantic

TABLE III
AVERAGE CLASSIFICATION ACCURACY OF THE VGG-16 MODEL FINE-TUNED WITH DIFFERENT DATA VOLUMES.

Models	Classification Accuracy
VGG-16 (fine-tune with FACD only)	0.8448
VGG-16 (fine-tune with the entire dataset)	0.9565

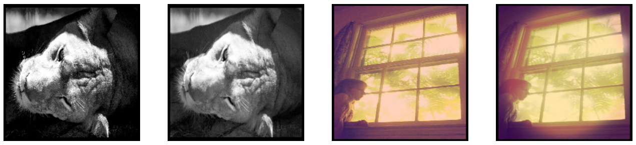
TABLE IV
AVERAGE CLASSIFICATION ACCURACY OF DIFFERENT SEMANTIC CATEGORIES, BASED ON THE VGG-16 MODEL.

	animal	architecture	cityscape	flora	food&drink	landscape	portrait	still life
Accuracy	0.9317	0.9305	0.9318	0.9094	0.9154	0.9313	0.9267	0.9244



(a) Brannan (b) LordKevin (c) Nashville (d) Poprocket

Fig. 2. Four correct classification results. The filters from left to right: Brannan, LordKevin, Nashville, and Poprocket.



(a) Label: Inkwell; Classified as: Willow (b) Label: Willow (c) Label: LordKevin; Classified as: Toaster (d) Label: Toaster

Fig. 3. Misclassification cases and the corresponding filtered photos.

categories is similar, which demonstrates robustness of the transformed models.

Figure 2 shows four correct classification results given by the VGG-16 model. For a randomly given photo, it is difficult for human beings to discriminate or classify the filter applied to it. However, by the deep models with appropriate fine-tuning, very high classification accuracy can be achieved.

Figure 3 shows two misclassification results. Figure 3(a) is classified as the Willow filter, while it actually was applied by the Inkwell filter. Figure 3(b) shows the lion photo really applied with the Willow filter. We see that visual effects of Figure 3(a) and Figure 3(b) are quite similar. It is definitely not a trivial task for human beings to discriminate Inkwell and Willow. Similarly, Figure 3(c) is classified as the Toaster filter, while it actually was applied with the LordKevin filter. Figure 3(d), which is really applied with the Toaster filter, is quite similar to Figure 3(c). These samples show the challenge of photo filter classification.

B. Filter Recommendation

We evaluate performance of filter recommendation based on the FACD dataset [2]. This dataset includes 1,280 reference (unfiltered) images, and through the manual pairwise comparison scheme designed in [2], averagely 3.7 filter types are recommended to each image. Notice that different users would prefer different types of filters for the same image, and thus recommending multiple filters to an image is not surprising. Following the evaluation setting in [2], we respectively recommend top 1, top 3, and top 5 filters to each test image, and evaluate the success rate (accuracy) of recommendation.

Table V shows average accuracy of filter recommendation based on different methods and different visual descriptors. Notice that in [2], 160 of the 1,280 images were selected as the testing data, and the remaining ones were taken as training data. However, we don't exactly know which 160 images were selected. In our work, we take all 1,280 images as the test images and calculate the average accuracy. Therefore, the comparison between our methods and others shown in Table V are not completely fair. We call the set of results based on this setting as "mismatch cases" and show them in the second section of Table V. We can observe some performance trends. First, our full model (Auto+Place+NIMA+YOLO) in the mismatched cases slightly outperforms [17] in terms of top 3 and top 5 accuracies. We may get more performance gain if more features are integrated in the future. Second, in these mismatch cases, the best performance is still inferior to the state of the arts [2]. In our work, the matrix for factorization and recommendation is constructed based on our collected images, and images in the FACD dataset are totally "unseen" to our model. The gap of data characteristics may cause performance drop. On the other hand, we don't require manually labeled user preference in the training data. Our methods are therefore easy to be extended when more training data are available.

To check the influence of the gap of data characteristics mentioned above, we further try taking training data and testing data only from the FACD dataset. Following the setting mentioned in [2], we randomly select 160 of the 1,280 images as the testing data, and the remaining ones are taken as training data. We run training and testing for five times, and report the average accuracies. We call this setting as "match cases". The last section of Table V shows that our full model (Auto+Place+NIMA+YOLO) achieves performance quite close to the state of the art. Our model ties with [2] at top 1 accuracy, is inferior to [2] at top 3 accuracy,

TABLE V
AVERAGE ACCURACY OF FILTER RECOMMENDATION BASED ON DIFFERENT METHODS AND DIFFERENT VISUAL DESCRIPTORS.

Methods	Top 1	Top 3	Top 5
Random Guess	16.80%	-	-
AlexNet [8]	33.13%	70.63%	88.75%
RAPID net [17]	37.50%	72.50%	86.25%
PairComp+Cate (AlexNet) [2]	41.25%	**80.00%**	89.18%
PairComp+Cate (RAPID net) [2]	**41.88%**	79.50%	90.00%
Mismatch cases			
Auto. features	33.58%	71.14%	85.48%
Place	31.76%	71.14%	83.48%
NIMA	31.40%	70.50%	80.23%
YOLO	27.77%	65.00%	80.45%
Auto + Place + NIMA + YOLO	35.23%	74.38%	87.73%
Match cases			
Auto. features	35.63%	66.87%	87.50%
Place	41.25%	68.13%	84.38%
NIMA	33.75%	71.25%	83.75%
YOLO	26.13%	70.00%	86.25%
Auto + Place + NIMA + YOLO	**41.88%**	76.25%	**91.87%**

but is superior to [2] at top 5 accuracy. Given that the proposed method does not require manual labeling, we think our model has much potential for future studies.

VI. CONCLUSION

In this paper, we present employing transfer learning to transform neural networks pre-trained for object classification into photo filter classification. We then build a filter recommender system based on the filter classifier. The contributions of this paper are threefold. First, we collect a dataset specifically for photo filter classification. Second, we comprehensively investigate the effectiveness of transfer learning, including performance of different models, performance variations yielded by different volumes of training data, and performance variations of different semantic categories. Third, we build an easy-to-extend filter recommender system without the requirement of manually labeled user preference. We believe that the obtained encouraging performance gives good foundation for future filter-related researches.

ACKNOWLEDGMENT

This work was partially supported by the Ministry of Science and Technology under the grant 108-2221-E-006-227-MY3, 107-2221-E-006-239-MY2, 107-2923-E-194-003-MY3, 107-2627-H-155-001, and 107-2218-E-002-055.

REFERENCES

[1] S. Bakhshi, D.A. Shamma, L. Kennedy, and E. Gilbert, "Why we filter our photos and how it impacts engagement," in *Proceedings of International AAAI Conference on Web and Social Media*, 2015, pp. 12–21.

[2] W.-T. Sun, T.-H. Chao, Y.-H. Kuo, and W.H. Hsu, "Photo filter recommendation by category-aware aesthetic learning," *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1870–1880, 2017.

[3] Z. Wu, Z. Sun, T. Kim, M. Reani, C. Jay, and X. Ma, "Mediating color filter exploration with color theme semantics derived from social curation data," in *Proceedings of the ACM on Human-Computer Interaction*, 2018, vol. 2.

[4] A.G. Reece and C.M. Danforth, "Instagram photos reveal predictive markers of depression," *EPJ Data Science*, vol. 16, no. 5, 2017.

[5] N. Murray, L. Marchesotti, and F. Perronnin, "Ava: A large-scale database for aesthetic visual analysis," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2408–2415.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C.L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of European Conference on Computer Vision*, 2014, pp. 740–755.

[8] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of International Conference on Learning Representations*, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.

[11] Simone Bianco, Claudio Cusano, and Raimondo Schettini, "Artistic photo filtering recognition using cnns," in *Proceedings of International Workshop on Computational Color Imaging*, 2017, pp. 249–258.

[12] Yu-Hsiu Chen, Ting-Hsuan Chao, Sheng-Yi Bai, Yen-Liang Lin, Wen-Chin Chen, and Winston H. Hsu, "Filter-invariant image classification on social media photos," in *Proceedings of ACM International Conference on Multimedia*, 2015, pp. 855–858.

[13] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[14] M.W. Berry, M. Browne, A.N. Langville, V. Paul Pauca, and R.J.Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics and Data Analysis*, vol. 52, no. 1, pp. 155–172, 2007.

[15] Z. Zheng, J. Yang, and Y. Zhu, "Nima: Neural image assessment," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 101–110, 2007.

[16] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[17] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, "Rapid: Rating pictorial aesthetics using deep learning," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 457–466.

[18] Hossein Talebi and Peyman Milanfar, "Initialization enhancer for non-negative matrix factorization," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3998–4011, 2018.

[19] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *https://arxiv.org/abs/1804.02767*, 2018.