Color Naming for Multi-Color Fashion Items

Vacit Oguz Yazici[1,2], Joost van de Weijer[1], and Arnau Ramisa[2]

[1] Computer Vision Center, Universitat Autonoma de Barcelona, Building O Campus UAB, 08193 Bellaterra, Barcelona, Spain
{voyazici}@cvc.uab.cat
{joost}@cvc.uab.cat
[2] Wide Eyes Technologies, Barcelona, Spain
{aramisa}@wide-eyes.it

Abstract. There exists a significant amount of research on color naming of single colored objects. However in reality many fashion objects consist of multiple colors. Currently, searching in fashion datasets for multi-colored objects can be a laborious task. Therefore, in this paper we focus on color naming for images with multi-color fashion items. We collect a dataset, which consists of images which may have from one up to four colors. We annotate the images with the 11 basic colors of the English language. We experiment with several designs for deep neural networks with different losses. We show that explicitly estimating the number of colors in the fashion item leads to improved results.

Keywords: deep learning, color, multi-label

1 Introduction

Computer vision offers great potential to develop tools to improve interaction between buyers and sellers in the fashion industry [1,2,3]. Color attributes (in this article referred to as color names) are among the essential properties of fashion items and their understanding is therefore crucial for efficient interaction with users. Therefore, in this article we focus on the automatic estimation of color names of images of fashion items. We will focus on extracting the colors of the fashion items in real-world images with background clutter and without available segmentation masks or bounding boxes which indicate the exact location of the fashion item. The task therefore is twofold, automatic detection of the fashion item, and estimation of its colors.

Color naming is a challenging task due to several reasons, including discrepancies between the physical nature of color and human perception (which is also affected by the cultural context), or external factors like varying illumination and complex backgrounds. Moreover complex background, human skin, or human hair act as clutter that deteriorates the accuracy of models. It is important to minimize the effects of this type of clutter in order to improve accuracy. A further difficulty of color naming in fashion, which is the focus of this paper, is that many of the objects that we see in the real world have several colors, which complicates the decision making process for algorithms.

Computational color naming has primarily focused on the 11 basic colors of the English language [4,5]. Those 11 basic colors are defined in the seminal work of Berlin and Kay [6] in which they researched the usage of color names in various different languages. Color names have been successfully used in a number of computer vision applications, including action recognition, visual tracking and image classification; see [7] for an overview. In the field of fashion image understanding, Liu et al. [8] do color naming using Markov Random Fields to infer category and color labels for each pixel in fashion images. To the best of our knowledge, all existing work on color naming focuses either on single colored objects or pixel-wise predictions.

Therefore, we address the problem of color name assignment to multi-color fashion items. We design several neural network architectures and experiment with various loss functions. We collect our own multi-label color dataset by crawling data from Internet sources. We show that a network with an additional classification head that explicitly estimates the number of color names improves performance. In addition, we show in a human annotation experiment that multi-color naming is an ambiguous task and human annotation results are only a few percent higher than results obtained by our best network.

The rest of this paper is organized as follows. Related work is discussed in Section 2. Section 3 describes details of the dataset that we use for the experiments. Section 4 elaborates the proposed approach. Experiments are presented in Section 5. Finally, we conclude this paper in Section 6.

2 Related Work

Research papers for fashion firstly focused on the segmentation of fashion products in images. Yamaguchi et al. [1], propose the Fashionista dataset consisting of 158,235 fashion photos with associated text annotations. They use a Conditional Random Field Model (CRF) in order to parse fashion clothes pixel-wise. However, their algorithms require fashion tags during the test time to get good accuracies. Simo-Serra et al. [2] address this issue and also propose a CRF model that exploits different image features such as appearance, figure/ground segmentation, shape and location priors for cloth parsing. They manage to obtain state-of-the-art performance on the Fashionista dataset. Liu et al. [9] propose a novel dataset which consists of 800,000 images with 50 categories, 1,000 descriptive attributes, bounding box and clothing landmarks. Moreover, they also propose a novel neural network architecture which is called FashionNet. The network learns clothing features by jointly predicting clothing attributes and landmarks. They do pooling and gating of feature maps upon estimated landmark locations to alleviate the effect of clothing deformation and occlusion. Recently, Cervantes et al. [3] propose a hierarchical method for the detection of fashion items in images.

For color, Cheng et al. [10] use a modified version of VGG for pixel-wise prediction out of 11 color labels (which are blue, brown, gray, white, red, green, pink, black, yellow, purple and orange) and a CRF to smooth the prediction. Although

Fig. 1: Sample images from the dataset with varying content and background clutter. Note that we do not provide segmentation and therefore to estimate the colors, the algorithm needs to implicitly segment the main fashion item.

their model is robust to background clutter, and can produce pixel-wise prediction, it is not robust to other clutter such as skin and hair color. Van de Weijer et al. [4], use probabilistic latent semantic analysis (PLSA) on Lab histograms to learn color names. Benavente et al. [5], present a model for pixel-wise color name prediction by using chromaticity distribution. Wang et al. [11], propose an algorithm which has two stages: in the first stage, which they name self supervised training, they train a shallow network with color histograms of random patches from the dataset. In the second stage, they fine-tune the same network to predict 11 basic colors. Mylonas et al. [12], use a mixture of Gaussian distributions. Schuerte and Fink [13] propose a randomized hue-saturation-lightness (HSL) transformation to get more natural color distributions; secondly, they used probabilistic ranking to remove the outliers. They claim that these steps helps color models accommodate to the variances seen in real-world images. In none of the before mentioned works to task of color naming multi-color objects is addressed.

3 Multi-Color Name Dataset

There are several datasets for color name learning. Van de Weijer et al. [4] introduced two datasets, constituted of images of objects retrieved from Google and EBAY respectively, and labeled with the 11 basic color names. Liu et al. [8] introduce another dataset which consists of 2682 images with pixel-level color annotations of the 11 basic colors plus a "background" class. However, almost every image in the dataset has a single color. To the best of our knowledge there is no dataset which explicitly considers multi-color objects.

We therefore collect a new dataset for this article, composed of images of fashion objects with one to nine colors (see Table 1). Single colored fashion images are crawled from various online shopping sites, and most of the multicolor labeled images are obtained by querying the Google images search engine with a query term containing a pair of color names and a fashion keyword (e.g. red and blue skirt) and downloading the 100 first images. There are 67 fashion keywords that we use and 55 color pairs that can be obtained with combinations of 11 basic

Table 1: The number of images for each color category

	one	two	three	four	five	six	seven	eight	nine	Total
Train	5556	5431	2178	1203	476	131	19	4	3	15001
Test	50	50	50	50	0	0	0	0	0	200

colors. At the end, we remove irrelevant and noisy data and crop the fashion item to prepare the dataset.

This process allows us to obtain images with two colors and more colors, as sometimes the search engine also returns images with additional colors not included in the query. Unfortunately, this leads to an imbalance between the number of 2-colored images and multi-colored images. Directly crawling for products with more than two colors using Google Images produces unsatisfactory results.

The dataset includes different types of images of varying complexity: catalog shots with smooth or complex background, images with plain background without any person or images taken by social media users; all labeled with the color names of the main fashion item. Sample images from the dataset can be seen in Figure 1. It should be noted that we do not use segmentation for the images, and naming the multiple colors of the fashion items includes dealing with clutter from the background, occlusions, and skin and hair of the person. However, if there is more than one fashion item in an image, to avoid any confusion, we provide a bounding box for the correspondent fashion item. In any case, the network has to implicitly segment the fashion item from occlusions and clutters.

4 Networks for Multi-Color Name Prediction

Methods on color naming focus on single colored objects. In this work we aim to propose a method for multi-colored fashion items. We evaluate several network architectures and losses for this task.

4.1 Network Design

In principle we believe the mapping from RGB to color names not to be highly complicated and only several layers are required. However, differentiating background from foreground is a highly complex process that requires many layers and should be implicitly done by the network.

First, we propose a shallow network; the truncated version of Alexnet [14]. We keep the first five convolution layers of the architecture and remove the fully connected layers of 4096 dimension. At the end we add a fully connected layer which maps features to the eleven basic color names. As a second network we use the full Alexnet architecture. Both nets are initialized with pretrained weights from ILSVRC 2012 dataset [15]. We think that finetuning from this model can alleviate noise caused by clutter such as complex backgrounds, hair or skin.

4.2 Loss Functions

We consider two loss functions for the purpose of color naming for multi-color fashion items. The first loss we consider to train the network is the softmax cross-entropy loss (SCE). The softmax cross-entropy can be seen in Equation 1.

$$L_{sce} = -\frac{1}{N}\sum_{i}^{N} P(i)\log Q(i) \tag{1}$$

where Q the predicted color distribution, P is the true color distribution, and N is the number of images. Q is obtained by applying a softmax normalization to the output of the last fully connected layer of the network, and the ground truth P is computed by assigning a uniform probability to all color names annotated for the fashion item (e.g. in case of three annotated color names, P would contain three elements with value 0.33).

While the softmax cross-entropy loss teaches a network to compute color probability distributions for an input fashion item, no decision is made on the actual number of colors. To remedy this, a threshold on the computed probabilities Q, learned from an independent validation set, is used to discard the colors unlikely to be really present.

The second loss we consider is the binary cross-entropy loss (BCE), which inherently supports multi-label classification. This loss is commonly used for attribute detection [16,9] because it models the presence of multiple labels simultaneously. Therefore it is expected to obtain better results than the softmax-cross entropy. Unlike with the softmax cross-entropy loss, the computed probability for a color name is independent of the others. For example the probability of both 'green' and 'orange' can be one simultaneously, something which is impossible for the softmax cross-entropy loss. Therefore, the loss trains 11 binary classifiers for each color. In Equation 2, the binary cross-entropy can be seen.

$$L_{bce} = -\frac{1}{N}\sum_{i}^{N} P_i \log Q_i + (1-P_i)\log(1-Q_i) \tag{2}$$

Similarly as the softmax-cross entropy loss we determine a threshold on a validation set to decide on the colors which are present in the fashion item. We found this to yield better results than choosing the natural threshold of 0.5.

4.3 Extra Head to Explicitly Estimate Number of Colors

In the previous section we consider two losses to estimate the color names. In principle the binary cross-entropy loss which implements the multi-label softmax cross-entropy loss is more suitable for the estimation of multiple colors. However, the probabilities which are the outcome of these networks both encode information of the number of color names as well as the confidence of the network in its estimation of the color names. Considering a single colored object which the system is not sure to label with either 'orange' or 'red', the algorithm that is

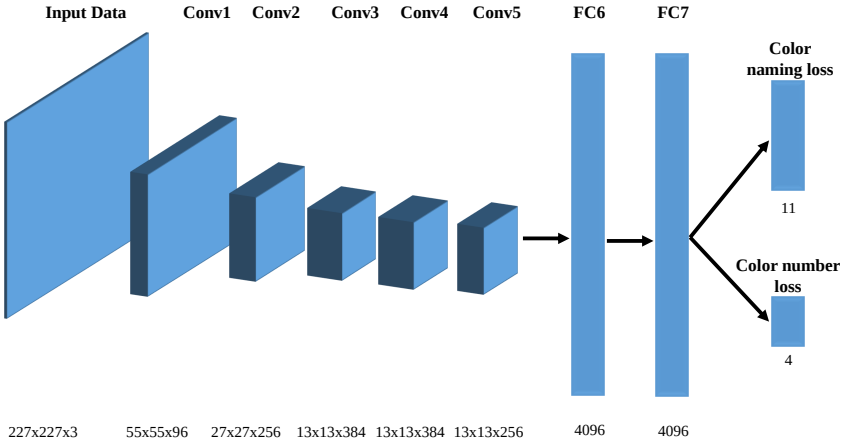

Fig. 2: The architecture of the deep network with the extra head.

based on binary cross-entropy might give both colors a probability of 0.6. Based on this we might conclude that the object is a multi-color object which is both 'orange' and 'red'. However, looking at the object it might be obvious that it only has a single color.

Therefore, we experiment with adding an extra classification head to the network, which explicitly estimates the number of colors in the main object. We model this objective as a classification task, and define four possible classes: one, two, three and four or more colors. A natural choice for this objective is the softmax cross-entropy loss layer, typically used for classification. In the experiments, we add this additional objective both to the networks which use softmax-cross entropy loss and the binary cross-entropy loss. The architecture of the network can be seen in Figure 2.

4.4 Training procedure

To train the network, we finetune from an Alexnet model which is trained on ILSVRC 2012 [15] using the Caffe framework [17]. The batch size is 64, the optimization method is SGD with momentum, set to 0.99, and we decrease the learning rate after every 5000 iterations. The initial learning rate is 0.0001 and the maximum iteration number is 20000. We also use data augmentation techniques in order to increase the accuracy of the models. The data augmentation techniques that we use are changing contrast, rescaling image and cropping random parts from images. Rescaling basically consists on changing the resolution of the image before resizing to the required network input size. The probability that any augmentation technique is applied to an image is 50%. We never keep

Table 2: Results of our models and the human annotators

		Shallow BCE w/ extra head	SCE w/o extra head	SCE w/ extra head	BCE w/o extra head	BCE w/ extra head	Human Score
micro	precision	77.24%	85.64%	80.27%	82.31%	83.57%	81.39%
	recall	67.20%	63.20%	70.80%	69.80%	71.20%	81.91%
	F1	71.87%	72.73%	75.24%	75.54%	**76.89%**	81.32%
macro	precision	78.40%	84.73%	79.82%	81.41%	83.10%	81.60%
	recall	65.24%	62.10%	69.78%	67.43%	69.33%	80.89%
	F1	69.32%	69.38%	73.46%	72.51%	**74.19%**	80.18%

both the original and the augmented image in the same batch, as we have observed that it may negatively impact the accuracy of the learned model. Finally, to avoid aspect ratio distortions caused by the resizing process, we use a padding function in order to make all images square.

5 Experiments

To evaluate the performance we use label based metric methods. We calculate the micro-precision, micro-recall, micro-F1, macro-precision, macro-recall and macro-F1. In the micro methods we sum up true positive, false positive and true negative for each label in order to get micro-recall and micro-precision. In the macro methods, we calculate precision and recall of each label and average them in order to get the macro-recall and macro-precision. The main difference is that the macro metrics do not take the label imbalance into account. To clarify the difference between the micro and macro methods, here we give the micro-precision and macro-precision:

$$P_{micro} = \frac{\sum_{j=i}^{L} tp_j}{\sum_{j=0}^{L} tp_j + \sum_{j=0}^{L} fp_j} \qquad P_{macro} = \sum_{j=0}^{L} \frac{tp_j}{tp_j + fp_j} \tag{3}$$

L is the number of classes, tp_j and fp_j is the true positive and false positive of class j. All of the results can be seen in Table 2. We focus on the F1-score which is a fair metric to compare methods. We first evaluate the two network architectures, namely the shallow and deep network. Both of the models have the extra head which forces them to learn the number of colors on a fashion item. The deep model clearly outperforms the shallow model. We attribute this to the fact that the shallow model is not able to segment the fashion items implicitly, and therefore fails for the more cluttered cases as can be seen in Figure 3.

Next we evaluate the different losses, and we verify if the additional head which explicitly predicts the number of colors contributes to a performance gain. It can be seen that adding the additional objective improves both the softmax cross-entropy loss and the binary cross-entropy loss; it forces the network to

(a) **GT**: **DE**: **SH**: (b) **GT**: **DE**: **SH**: (c) **GT**: **DE**: **SH**:

Fig. 3: Qualitative results of the shallow and deep networks. GT, DE, SH denote the ground truth, the predictions of the deep and shallow model respectively. Note that the networks should estimate the colors of the fashion item while ignoring the non-relevant colors present in the background.

learn the number of colors on a fashion item, and also contributes to name them as can be seen when comparing the columns 4-5 and 6-7 of Table 2. During the inference, the extra head can predict maximum 4 colors. In case the networks without extra head predicts more than 4 colors, we get the first 4 with the highest scores.

In the last column of Table 2, we show the average performance obtained by humans for the same task. We asked seven annotators of different ages and backgrounds to provide labels for the images in the test set. The obtained scores for humans show that multi-color labelling is an ambiguous task, and for many objects humans do not agree on the labels. This score can be considered to be an upper bound for computational methods.

The contribution of adding an extra head is shown in Figure 4. From the ground truth and prediction of the cross entropy models it can be seen that the extra head provides robustness if the color distribution is not uniform in the image. However, it makes the model more conservative and biases it towards predicting a lower number of colors in the image (last two examples on the right).

6 Conclusions

In this paper we address the problem of color name estimation in multi-colored objects that, to the best of our knowledge, we are the first to address. We collect a dataset of over 15.000 images with a varying number of colors per object and we evaluate several network architectures for the purpose of multi-color estimation. Preliminary results show that adding an additional objective to explicitly estimate the number of colors in the object improves results. Following recent work we are interested in extending the set of color names to include a wider

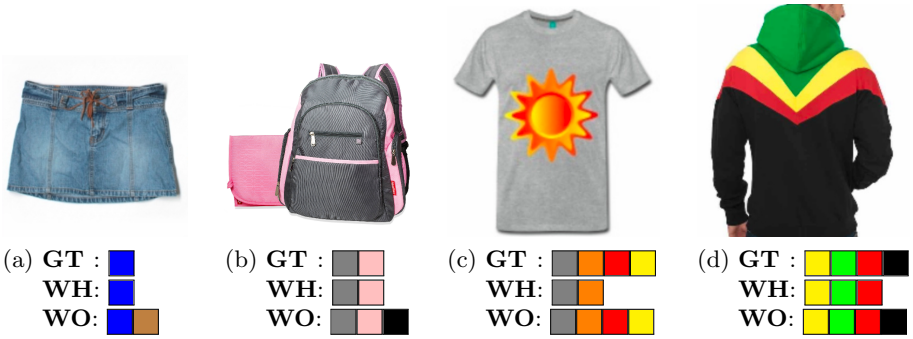

Fig. 4: Qualitative results of the BCE model with (WH) and without (WO) the extra head.

range of colors [18,19]. We hope that this paper further motivates researchers to investigate the more realistic setting of color naming for multi-colored objects.

Acknowledgements

This work was supported by TIN2016-79717-R of the Spanish Ministry and the CERCA Programme and the Industrial Doctorate Grant 2016 DI 039 of the Ministry of Economy and Knowledge of the Generalitat de Catalunya.

References

1. Yamaguchi, K., Kiapour, M.H., Ortiz, L.E., Berg, T.L.: Parsing clothing in fashion photographs. In: CVPR, IEEE (2012) 3570–3577
2. Simo-Serra, E., Fidler, S., Moreno-Noguer, F., Urtasun, R.: A High Performance CRF Model for Clothes Parsing. In: ACCV. (2014)
3. Cervantes, E., Yu, L.L., Bagdanov, A.D., Masana, M., van de Weijer, J.: Hierarchical part detection with deep neural networks. In: ICIP. (2016) 1933–1937
4. Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Transactions on Image Processing **18**(7) (2009)
5. Benavente, R., Vanrell, M., Baldrich, R.: Parametric fuzzy sets for automatic color naming. JOSA A **25**(10) (2008) 2582–2593
6. Berlin, B., Kay, P.: Basic color terms: their universality and evolution. Berkeley: University of California (1969)
7. Van De Weijer, J., Khan, F.S.: An overview of color name applications in computer vision. In: CCIW. (2015) 16–22
8. Liu, S., Feng, J., Domokos, C., Xu, H., Huang, J., Hu, Z., Yan, S.: Fashion parsing with weak color-category labels. IEEE Transactions on Multimedia **16**(1) (2014)
9. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: CVPR. (2016)
10. Cheng, Z., Li, X., Loy, C.C.: Pedestrian color naming via convolutional neural network. In: ACCV, Springer (2016) 35–51

11. Wang, Y., Liu, J., Wang, J., Li, Y., Lu, H.: Color names learning using convolutional neural networks. In: ICIP, IEEE (2015) 217–221
12. Mylonas, D., MacDonald, L., Wuerger, S.: Towards an online color naming model. In: Color and Imaging Conference. Volume 2010., Society for Imaging Science and Technology (2010) 140–144
13. Schauerte, B., Fink, G.A.: Web-based learning of naturalized color models for human-machine interaction. In: DICTA, IEEE (2010) 498–503
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1097–1105
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR, IEEE (2009) 248–255
16. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV. (2015) 3730–3738
17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM int. conf. on Multimedia, ACM (2014) 675–678
18. Mylonas, D., MacDonald, L.: Augmenting basic colour terms in english. Color Research & Application (2015)
19. Yu, L., Zhang, L., van de Weijer, J., Khan, F.S., Cheng, Y., Parraga, C.A.: Beyond eleven color names for image understanding. Machine Vision and Applications (2018)