CoRe: Color Regression for Multicolor Fashion Garments

Alexandre Ramé[1,2], Arthur Douillard[1,2], Charles Ollion[2]
[1]Sorbonne Université, [2]Heuritech,

Abstract

Developing deep networks that analyze fashion garments has many real-world applications. Among all fashion attributes, color is one of the most important yet challenging to detect. Existing approaches are classification-based and thus cannot go beyond the list of discrete predefined color names. In this paper, we handle color detection as a regression problem to predict the exact RGB values. That's why in addition to a first color classifier, we include a second regression stage for refinement in our newly proposed architecture. This second step combines two attention models: the first depends on the type of clothing, the second depends on the color previously detected by the classifier. Our final prediction is the weighted spatial pooling over the image pixels RGB values, where the illumination has been corrected. This architecture is modular and easily expanded to detect the RGBs of all colors in a multicolor garment. In our experiments, we show the benefits of each component of our architecture.

1. Introduction

Convolutional Neural Networks (CNNs) [20, 26] generated a lot of interest in the fashion industry. Recent datasets of fashion images [17, 28, 51] encouraged various approaches for attributes classification [7, 41], visual search [13, 19, 30, 45] and object detection [23, 34].

One of the key attributes to describe a fashion item is its color. However, colors are subjective properties of garments as not all humans recognize colors the same way [32]: their automatic estimation is therefore challenging. So far, most approaches consider the problem through the angle of **discrete color naming**: a classifier chooses amongst 11 colors of the English language [4], based on features from histograms [2, 9, 40] or from CNNs [10, 21, 24, 33, 43, 46, 47]. Recent approaches increased the number of color names up to 28 [48] or 313 [50]. We take a step forward and tackle the problem through the angle of continuous color regression. Rather than only predicting an approximate discrete color name, we aim at predicting the exact **continuous color RGB**. This refined information is necessary for many industrial applications such as precise visual search and fine-grained trend detection to handle color inherent ambiguity. Moreover, the regression paradigm is more versatile and adaptable; this is especially important for real-world applications, where product requirements can evolve. In contrast, adding a new color name in classification approaches would require a dataset relabelling and the training of a new network.

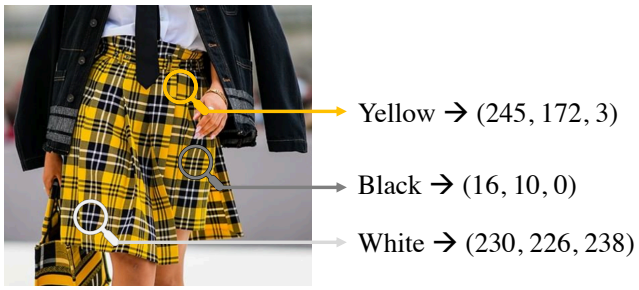

Figure 1. **Two-stage color regression**. After detecting the main color names of the clothing (the skirt here), our architecture regresses the exact and refined RGB values.

To understand the challenge of this regression task, let's consider an unsupervised approach, that would naively average the different pixel RGBs of the image. That would fail for three reasons. *First*, it would suffer from varying illuminations, sparking discrepancies between the raw pixels and what a human would perceive in ideal white light. *Second*, it would consider 'parasite' pixels from either the background or other garments. *Third*, in case of multicolor garments, it would average pixels belonging to different colors and would predict neutral RGBs.

In this paper, we propose a new supervised two-stage architecture for color regression, inspired by the object detection literature [35]. The first stage is a standard color classifier — similar to previous works [10, 33, 43, 46, 47] — which predicts the main colors of the considered garment. Our contributions lie in the second regression-based stage, refining the previous discrete prediction by weighting the different pixels of the illumination-corrected image.

First, we developed an illumination module that auto-

matically corrects the contrast of the image — and this without supervision. This is inspired by the color constancy literature [3, 5, 6, 15, 22, 29, 37, 38]. *Second*, in order to reduce the impact of complex backgrounds or complex garments structure, we leverage a semantic segmenter [10, 44] pretrained to detect fashion garments. *Third*, we detect all pixels in this garment that are close to the colors predicted by the first stage. This enables the handling of multicolor garments by removing pixels from conflicting colors. These last two components are combined to find the appropriate weighting of each pixel in the image — enabling to focus on the appropriate regions of the image.

Empirically, we validate the effectiveness of our approach in our new fashion dataset collected from real-world images. More importantly, we show that all our components contribute.

2. Model

The network in Figure 2 is trained end-to-end on the task of continuous color regression, with annotated RGB values, but with neither illuminations nor clothing segmentation ground-truth annotations.

2.1. Pixels correction

Our model needs to correct the bias in the pixels of the image, for example the quality or the lighting of the photo. To do so, first, all images are **contrast normalized** in a preprocessing step by global histogram stretching in order to be robust across various lighting conditions. Note that this does not change the relative contributions of the three RGB channels. Second, to remove **illumination** color casts, we need to detect the initial illumination of the image (a scalar value per channel); this is a regression task, achieved easily by a deep neural network. Note that following [29], we choose explicitly the *VGG16* [39] neural architecture because of its capacity to extract low-level features [16]. Finally, each pixel in the image is corrected using the *Von Kries* method [42], *i.e.*, scaling each channel by the detected illumination value.

2.2. Spatial pooling over pixels for color regression

To sample only the relevant regions of the image, we apply a spatial pooling over the pixels of the image weighted by two complementary attention module. They modulate the importance of each pixel in the spatial pooling. The first **object-attention** focuses on the clothing category, the second **colorname-attention** focuses on the color name predicted. These attention masks are then multiplied pixelwise to create the **combined-attention**.

2.2.1 Through object-attention

Following the work from [47], we use the popular fully convolutional *Deeplabv3* [8]. This attention network is pretrained on the task of semantic segmentation over the clothing crops from *Modanet* [51] and therefore has an emphasis on the garment surface: it produces an **object-attention** different for each clothing category (top, coat, ...). This spatial prior is then fine-tuned: ideally, it would learn to identify the regions which contain the relevant color information given a garment type.

2.2.2 Through colorname-attention

The previous object-attention will fail for multicolor garments. Indeed, if the garment contains a set of distinct and distant RGB values, the mean of this set would be predicted: it leads to unsaturated color predictions. For example, in Figure 2, including the white pixels from the shirt in our spatial pooling would predict a RGB closer to light pink.

Predicting the RGB of the main color For simplicity, we first describe how to detect only the RGB of the main color of a (potentially multicolor) garment. *First*, we detect the main color name; specifically, following previous works, the *VGG16* features are followed by a fully connected layer with a softmax activation function that predicts a distribution over 72 color names. This is trained by minimizing a categorical cross-entropy loss $\mathcal{L}_{CE}$.

Second, our **colorname-attention** only selects pixels in the image sufficiently close to the color detected by this first color classifier. We map this discrete color to its $\mathcal{RGB}_c$ continuous value: *e.g.*, (134, 71, 71) for *velvet red*. Now, given a pixel p of RGB value $\mathcal{RGB}_p$, the colorname-attention CA_p for pixel p is:

$$CA_p \propto exp^{-\frac{1}{127.5^2 * \mathcal{T}}(\mathcal{RGB}_p - \mathcal{RGB}_c)^2}. \quad (1)$$

It sums to 1 over all pixels and its peakedness depends on $\mathcal{T}$, the temperature of the **RGB spatial softmax**. $\mathcal{T}$ is the only parameter of this colorname-attention module, which can be *either* a predefined hyperparameter, *either* learned, *either* input-dependant and predicted with a fully connected layer from VGG features. This last option works best as shown in our Experiments from Section 3.

Predicting the RGBs of all colors We can easily generalize our approach for detecting multiple RGBs. To do so, we predict multiple color names with a sigmoid [46] followed by a combination of categorical cross entropy and binary cross entropy as in [34]. Multiple colors share the same object-attention but have different colorname-attentions: therefore we predict different RGBs with our combined-attention.

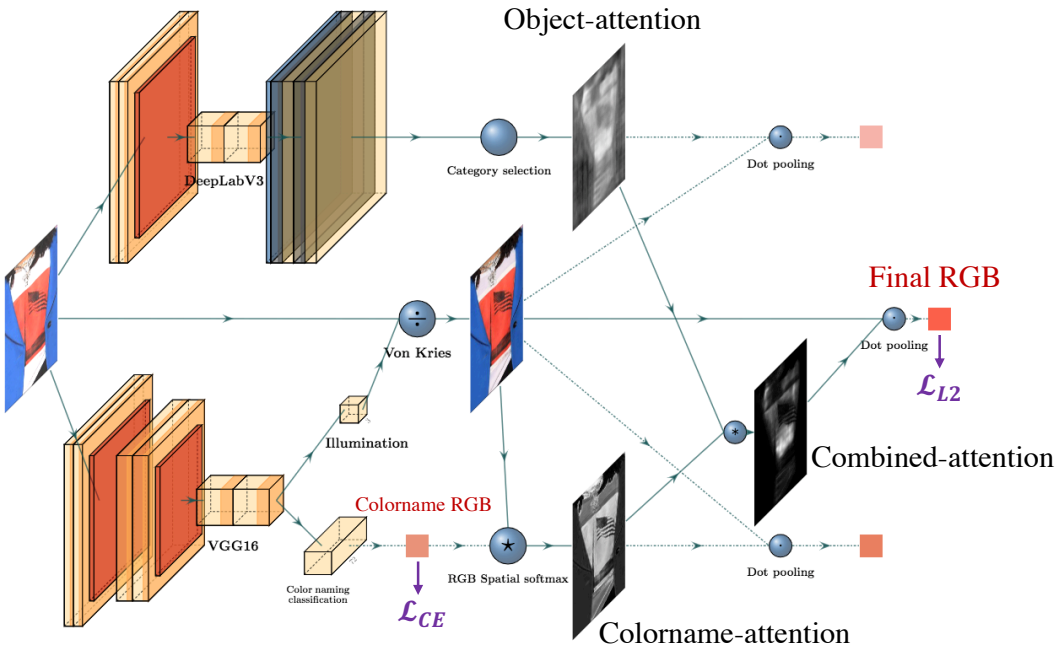

Figure 2. **Architecture components and training procedure**. *First*, the *DeepLabv3* **object-attention** network selects pixels inside the garment. *Second*, the *VGG16* CNN extract features before feeding two heads: the first predicts the image initial **illumination** (to be corrected with the *Von Kries* method), and the second head detects the **main discrete color name** amongst 72 available colors (trained with a categorical cross entropy $\mathcal{L}_{CE}$). This main color is used to create a **colorname-attention**. These attentions are used for weighted spatial pooling over the pixels of the image for **color regression**. Best results are obtained when colorname-attention and object-attention are combined, as this would select only pixels that are simultaneously inside the clothing and that have a RGB value close to the RGB of the predicted discrete color name. All components are trained so that the final RGB matches the RGB annotation with an Euclidean distance in the LAB space ($\mathcal{L}_{L2}$). This architecture is easily extended to detect the RGBs of multiple colors (red and white here).

The challenge is then to know how many colors should be predicted. Following the work from [46], we explicitly **learn the number of colors** in each garment (up to 3 different colors). Note that we apply class weights for handling unbalanced classes.

Selecting the color names with maximum scores would often predict several times the same major color: for example a *light red* and a *dark red* would converge towards a *medium red*. Thus, we apply a **non-maximum-suppression** algorithm to delete predicted RGBs too close to each other. In Figure 2, the network's confusion between two different kinds of reds would have prevented from predicting white.

2.3. Training overview

The LAB colorspace best models perceptual distance in colorization [11, 50]: thus, the chosen regression objective is $\mathcal{L}_{L2}$, the **Euclidean distance** between predicted and ground truth colors converted to LAB rather than RGB. Our two losses — color naming $\mathcal{L}_{CE}$ and color regression $\mathcal{L}_{L2}$ — are summed before backpropagation; this enables the learning of all architecture components end-to-end.

2.4. Discussion: analogy with Faster R-CNN

Our two-stage approach is highly inspired by the *Faster R-CNN* [35] architecture. The anchors of the Region Proposal Network are replaced by color names. In both cases, the first classifier gives a rough estimate, refined by the second regression stage. Specifically, given a selected anchor (resp. color name), the final regression adds a small continuous offset leading to a more precise box (resp. RGB values). Non-maximum-suppression algorithms are also standard to handle overlapping crops in object detection.

3. Experiments

3.1. Setup

Dataset As far as we know, there is no dataset for the task of continuous colors regression. Thus, we have collected a new dataset of 30,269 fashion garments: 5,363 coats, 8,166 dresses, 3,991 pants, 6,871 shoes and 5,878 tops. The validation dataset and the test dataset are composed of 2000 images: the other images are used for training. Each garment was labeled by a single operator with its exact color RGB: not like it appears in the image, but like the operator thought it would appear in ideal white light. The annotation process

Table 1. **Results for RGB regressions**: percentage of predictions closer to the ground truths RGBs than several thresholds (10, 20, 30, 40), according to the *deltaE ciede2000* distance [14, 31]. **Bold** highlights best score.

Method					Main color				All colors			
Name	Color-attention	Object-attention	Illumination	$\mathcal{T}$	≤ 10	≤ 20	≤ 30	≤ 40	≤ 10	≤ 20	≤ 30	≤ 40
Unsupervised K-means Clustering [27]					47	72	83	89	19	31	37	42
Colorname RGB [46]					51	72	87	92	34	52	64	68
Ours		✓		-	50	78	90	95	-	-	-	-
Ours	✓			Set to 1	48	76	88	93	35	56	65	68
Ours	✓	✓		Set to 1	59	87	93	95	44	64	69	71
Ours	✓	✓		Trainable	62	87	93	95	46	65	69	71
Ours	✓	✓	✓	Trainable	67	89	**94**	**96**	48	66	70	72
Our best	✓	✓	✓	Predicted	**73**	**90**	93	95	**54**	**68**	**71**	**73**

is therefore more complex and time-consuming than classical classification. The color names can automatically be derived by nearest neighbors in the *LAB* space. In case of multicolor items, the operators were asked to tag them in decreasing order of importance.

Implementation Our code is in Tensorflow [1] and Keras [12]. We chose an Adam [25] optimizer with a learning rate of 0.0001 and a batch of 16 images during 50 epochs. We applied standard data augmentation methods: random cropping and translation.

3.2. Results

Table 1 summarizes our experiments. To compare approaches, we count the percentage of predictions that are closer to the color annotation than a given threshold, leveraging the *deltaE ciede2000* distance [14, 31] that is arguably "the best metric for understanding how the human eye perceives color difference" [36]. With our best network that incorporates all our components, 73% of all predictions have a distance to the main color smaller than 10; in practice, these differences are small to the human eye. We also report similar metrics when the goal is to detect all colors in the garment. We showcase in Figure 3 multiple predictions of our model.

Figure 3. **Visualization of the predictions**. This picture illustrates predictions on two images. On the left image, the two interleaved garments (jacket and t-shirt) are well predicted despite their closeness thanks to our attention mechanism. In the right image, our model distinguishes the good yellow RGB despite the illumination and shadows that harden the task.

Baselines We compare our model against two existing baselines. First we train an unsupervised *K-means* pixels clustering [27] directly on the pixel space, after a background removal done by an external semantic segmentation model trained on Modanet [51]. Colors are ranked according to the number of pixels in each cluster. The second baseline, *Colorname RGB*, is the direct extension of the multitagger approach from [46]: it detects the color names (*e.g. velvet red*, *dark purple*, etc.) and then produces the associated RGB values. This baseline is depicted in Figure 2.

Attention First, fine-tuning the semantic segmentation model on the regression task (object-attention) already surpasses previous approaches when predicting only the main color. The colorname-attention improves results in the multi-color setup. These two attentions, when combined, are mutually reinforcing and complementary.

Temperature We show that the temperature $\mathcal{T}$ value is important: bigger $\mathcal{T}$ leads to sharp distribution and takes into account fewer pixels from the initial images that with a lower $\mathcal{T}$. Rather than grid-searching its optimal value, it can be learned for improved results. Moreover, the optimal $\mathcal{T}$ depends on the image: therefore, best results are obtained with $\mathcal{T}$ predicted from VGG features. This analysis is consistent with recent insights for calibration via input-dependant temperature scaling [49].

Illumination Finally, including the illumination module improves performances. Future work could further improve results by pretraining on the *Color Checker Dataset* [18].

4. Conclusion

In this paper we addressed the color regression problem for fashion garments. By collecting a unique dataset of 30,269 images, we empirically show the benefits of our newly proposed two-stage architecture. These performance gains would have a potential impact on many real-world usages, notably to better detect fashion trends and for visual search. Finally, we hope to shed light on properties of fashion garments (perhaps surprisingly) complex to detect.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. (p. 4).

[2] Nakhoon Baek, Sun-Mi Park, Ku-Jin Kim, and Seong-Bae Park. Vehicle color classification based on the support vector machine method. In *ICIP*, 2007. (p. 1).

[3] Jonathan T Barron. Convolutional color constancy. In *ICCV*, 2015. (p. 2).

[4] Brent Berlin and Paul Kay. *Basic color terms: Their universality and evolution*. Univ of California Press, 1991. (p. 1).

[5] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using cnns. In *CVPR*, 2015. (p. 2).

[6] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Single and multiple illuminant estimation using convolutional neural networks. *TIP*, 2017. (p. 2).

[7] Lukas Bossard, Matthias Dantone, Christian Leistner, Christian Wengert, Till Quack, and Luc Van Gool. Apparel classification with style. In *ACCV*, 2012. (p. 1).

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint*, 2017. (p. 2).

[9] Pan Chen, Xiang Bai, and Wenyu Liu. Vehicle color recognition on urban road by feature context. *T-ITS*, 2014. (p. 1).

[10] Zhiyi Cheng, Xiaoxiao Li, and Chen Change Loy. Pedestrian color naming via convolutional neural network. In *ACCV*, 2016. (p. 1, 2).

[11] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *ICCV*, 2015. (p. 3).

[12] François Chollet. Keras. https://github.com/fchollet/keras, 2015. (p. 4).

[13] Charles Corbiere, Hedi Ben-Younes, Alexandre Ramé, and Charles Ollion. Leveraging weakly annotated data for fashion image retrieval and label prediction. In *ICCV*, 2017. (p. 1).

[14] Mark D Fairchild. *Color appearance models*. John Wiley & Sons, 2013. (p. 4).

[15] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Trémeau, and Christian Wolf. Mixed pooling neural networks for color constancy. In *ICIP*, 2016. (p. 2).

[16] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, 2015. (p. 2).

[17] Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. *arXiv preprint*, 2019. (p. 1).

[18] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *CVPR*, 2008. (p. 4).

[19] M Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. Where to buy it: Matching street clothing photos in online shops. In *ICCV*, 2015. (p. 1).

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. (p. 1).

[21] Chris Hickey and Byoung-Tak Zhang. Hierarchical color learning in convolutional neural networks. In *CVPRW*, 2020. (p. 1).

[22] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc4: fully convolutional color constancy with confidence-weighted pooling. In *CVPR*, 2017. (p. 2).

[23] Menglin Jia, Yichen Zhou, Mengyun Shi, and Bharath Hariharan. A deep-learning-based fashion attributes detection model. *arXiv preprint*, 2018. (p. 1).

[24] K. R. Jyothi and M. Okade. Computational color naming for human-machine interaction. In *TENSYMP*, 2019. (p. 1).

[25] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. (p. 4).

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. (p. 1).

[27] Xander Lewis. Extracting colours from an image using k-means clustering. https://towardsdatascience.com/extracting-colours-from-an-image-using-k-means-clustering-9616348712be, 2018, (accessed Mars 9, 2022). (p. 4).

[28] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. (p. 1).

[29] Zhongyu Lou, Theo Gevers, Ninghang Hu, Marcel P Lucassen, et al. Color constancy by deep learning. (p. 2).

[30] Corey Lynch, Kamelia Aryafar, and Josh Attenberg. Images don't lie: Transferring deep visual semantic features to large-scale multimodal learning to rank. In *SIGKDD*. (p. 1).

[31] WS Mokrzycki and M Tatol. Colour difference e-a survey. (p. 4).

[32] Timea R Partos, Simon J Cropper, and David Rawlings. You don't see what i see: Individual differences in the perception of meaning from visual stimuli. *PloS one*, 2016. (p. 1).

[33] Reza Fuad Rachmadi and I Purnama. Vehicle color recognition using convolutional neural network. *arXiv preprint*, 2015. (p. 1).

[34] Alexandre Rame, Emilien Garreau, Hedi Ben-Younes, and Charles Ollion. Omnia faster r-cnn: Detection in the wild through dataset merging and soft distillation. *arXiv preprint*, 2018. (p. 1, 2).

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. (p. 1, 3).

[36] Zachary Schuessler. Delta e 101. http://zschuessler.github.io/DeltaE/learn/. [Online; accessed 10-Mars-2022]. (p. 4).

[37] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In *ECCV*. Springer, 2016. (p. 2).

[38] Oleksii Sidorov. Artificial color constancy via googlenet with angular loss function. *arXiv preprint*, 2018. (p. 2).

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. (p. 2).

[40] Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *TIP*, 2009. (p. 1).

[41] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *ICCV*, 2015. (p. 1).

[42] J von Kries. Chromatic adaptation, festschrift der alberchtludwig-universität, 1902. (p. 2).

[43] Yuhang Wang, Jing Liu, Jinqiao Wang, Yong Li, and Hanqing Lu. Color names learning using convolutional neural networks. In *ICIP*, 2015. (p. 1).

[44] Wojciech Wieclawek and Ewa Pietka. Car segmentation and colour recognition. In *MIXDES*, 2014. (p. 2).

[45] Kota Yamaguchi, M Hadi Kiapour, and Tamara L Berg. Paper doll parsing: Retrieving similar styles to parse clothing items. In *ICCV*, 2013. (p. 1).

[46] Vacit Oguz Yazici, Joost van de Weijer, and Arnau Ramisa. Color naming for multi-color fashion items. In *WorldCIST*, 2018. (p. 1, 2, 3, 4).

[47] Lu Yu, Yongmei Cheng, and Joost van de Weijer. Weakly supervised domain-specific color naming based on attention. In *ICPR*, 2018. (p. 1, 2).

[48] Lu Yu, Lichao Zhang, Joost van de Weijer, Fahad Shahbaz Khan, Yongmei Cheng, and C Alejandro Parraga. Beyond eleven color names for image understanding. *Machine Vision and Applications*, 2018. (p. 1).

[49] Jize Zhang, Bhavya Kailkhura, and T Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *ICML*, 2020. (p. 4).

[50] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. (p. 1, 3).

[51] Shuai Zheng, Fan Yang, M Hadi Kiapour, and Robinson Piramuthu. Modanet: A large-scale street fashion dataset with polygon annotations. *arXiv preprint*, 2018. (p. 1, 2, 4).