

Probabilistic Color Modelling of Clothing Items



Mohammed Al-Rawi and Joeran Beel

Abstract Color modelling and extraction is an important topic in fashion. It can help build a wide range of applications, for example, recommender systems, color-based retrieval, fashion design, etc. We aim to develop and test models that can extract the dominant colors of clothing and accessory items. The approach we propose has three stages: (1) Mask-RCNN to segment the clothing items, (2) cluster the colors into a predefined number of groups, and (3) combine the detected colors based on the hue scores and the probability of each score. We use Clothing Co-Parsing and ModaNet datasets for evaluation. We also scrape fashion images from the WWW and use our models to discover the fashion color trend. Subjectively, we were able to extract colors even when clothing items have multiple colors. Moreover, we are able to extract colors along with the probability of them appearing in clothes. The method can provide the color baseline drive for more advanced fashion systems.

Keywords Color clustering · Deep learning · Clothing · Fashion trends · k-Means · Gaussian mixture model

1 Introduction

Color modeling and extraction (Fig. 1) has long been an important topic in many areas of science, business and industry. Color is also one of the fundamental components of image understanding. Due to problems of color degradation over time and the possibility of having a large number of colors in every image, this area is still under extensive research [9, 18].

Color is one of the important cues that attracts customers when it comes to fashion. Fashion designers and retailers understand this and they usually make use of color services, such as the catalogs provided by Pantone (pantone.com). Moreover, fashion

M. Al-Rawi (✉) · J. Beel
ADAPT Centre, Trinity College Dublin, Dublin, Ireland
e-mail: alrawim@tcd.ie

J. Beel
e-mail: joeran.beel@tcd.ie

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
N. Dokoohaki et al. (eds.), *Recommender Systems in Fashion and Retail*,
Lecture Notes in Electrical Engineering 734,
https://doi.org/10.1007/978-3-030-66103-8_2



Fig. 1 Colors and their possibilities are extracted from the image shown to the left. Sweatshirt colors’ names are: ‘Egyptian blue’, ‘Rosso corsa’, ‘Old gold’, and ‘Spanish viridian’; pants color name is ‘USAFA blue’. Best viewed in color

and AI have found a common ground in recent years. In this regard, companies race to build products and services to serve customers. The impact of deep learning and the development of other AI methods provide companies with unprecedented means to achieve their goals. Amazon and StitchFix are providing their customers with the so called “Personal Styling Service,” which is semi-assisted by fashion stylists [7, 10]. Facebook is building a universal product understanding system where fashion is at its core [3, 4]. Zalando researchers proposed in [22] a model for finding pieces of clothing worn by a person in full-body or half-body images with neutral backgrounds. Some other companies are dedicated to build sizing and fitting services [14, 28, 32]. However, efforts dedicated for explicitly making use of color values in fashion AI are somehow limited. This is because most, if not all, products/works rely on the use of color tags. In addition, deep learning models, which are now the cutting edge technology used for several fashion AI apps, are color agnostic. That is, they do a remarkable prediction job without explicitly extracting the color values. This is because the color feature is implicitly extracted alongside other spacial features at the convolution layer.

Several works treated color as a classification problem. Color classification has been the topic of [27]. Other works used deep learning to predict one of eleven basic color names [29] and one of 28 color names [31]. Clearly, color classification has the disadvantage of dealing with a small number of tagged colors. This is a serious problem because of the failure to keep pace with the millions of colors that the human vision system can perceive [17]. This suggests that color extraction should be treated as a regression and not a classification problem.

There are a few works dedicated to extracting the main color values from images using regression methods. The authors of [23] presented a method for extracting color themes from images using a regression model trained on color themes that were annotated by people. To collect data for their work, the authors asked people to extract themes from a set of 40 images that consisting of 20 paintings and 20 photographs. However, such data-driven approach may suffer from generalization issues because millions of colors exist in the real world. The authors of [33] used k-means algorithm on an input image to generate a palette consisting of a small set of the most representative colors. An iterative palette partitioning based on cluster validation has been proposed in [18] to generate color palettes.

Clustering methods have been proposed to extract colors from images much earlier than classification and regression methods. Automatic palette extraction has been the focus of [12, 13] in which a hue histogram segmentation method has been used. The hue histogram segmentation has a disadvantage not only that it is affected by the saturation and intensity values, but also singularities when the saturation values are zero. Other works, as in [8], pointed out the advantages the fuzzy-c-means can provide over k-means even though they aimed at using it for color image segmentation. The main problem with clustering algorithms is knowing the number of colors a priori. Choosing a low value will result in incorrect color values if the image has more colors than the one used to build the clustering model. On the other hand, using a high number of clusters has the drawback of extracting several colors, some with approximate hue values. This makes it really difficult to extract colors accurately.

In this work we propose a multistage method intended for modeling and extracting color values from the clothing items that appear in images. Our pipeline makes use of Mask-RCNN [15], which is a deep neural network aimed to solve instance segmentation problem, to segment clothing items from each image. Next, we cluster the colors into a large number of groups; and then, merge the resultant colors according to their hue and probability values. We use a probabilistic model because it is possible for clustering algorithms to yield more than one color value for each pure color.

2 Color Modelling and Extraction

Stored as images, the colors of clothes usually suffer from severe distortions. The life of a clothing item, color printing quality, imaging geometry, amount of illumination, and even imaging devices affect how colors appear in images. Additionally, clothing

items differ in material, fabric, texture, print/paint technology, and patterns. Below, we introduce the mathematical intuition behind the color distribution, followed by a method for extracting colors from clothing items.

2.1 Mathematical Modelling

Without regard to complexity and color distortions, the single color and single channel image should have a Dirac delta density distribution defined as follows:

$$\delta(x|\mu, \sigma) = \lim_{\sigma \rightarrow 0} \frac{1}{\sqrt{\pi}|\sigma|} \exp\left(\frac{-(x - \mu)^2}{\sigma^2}\right), \quad (1)$$

where μ is the intensity value. RGB color images have three channels; hence, we write the density distribution of a single-color RGB image is given by:

$$\Delta(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \lim_{\Sigma \rightarrow 0} \frac{1}{\sqrt{\pi}|\Sigma|} \exp\left(-(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right), \quad (2)$$

where Σ is a $q \times q$ sized covariance matrix, q is the number of channels ($q = 3$ for an RGB image), $\boldsymbol{\mu}$ and \mathbf{x} are q sized vectors, and $|\Sigma|$ is the determinant of Σ . The distribution in (2) is a point in 3D space defined by the value $\boldsymbol{\mu}$. In the real world, and even when the imaging setting is typical, the color distribution of each channel will be normally distributed around μ . This can be denoted for the one channel single-color case as follows:

$$n(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right), \quad (3)$$

where σ parameter relates to color dispersion. For a single-color RGB image, the density is a multivariate Gaussian distribution given by:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi}|\Sigma|} \exp\left(-(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right). \quad (4)$$

And for a multicolor image, we model the colors as a Gaussian mixture model prior distribution on the vector of estimates, which is given by:

$$p(\mathbf{x}) = \sum_{i=1}^K \phi_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \Sigma_i), \quad (5)$$

where

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{(2\pi)^K |\boldsymbol{\Sigma}_i|}} \exp\left(-(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\right), \quad (6)$$

K denotes the number of colors, and the i_{th} vector component is characterized by normal distributions with weight ϕ_i , mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$.

Expressed as a Gaussian mixture model, the color distribution also becomes extremely complicated when the clothing item has more than one color. However, the distribution may become more complicated because there is a possibility of frequency deviation of the colors during the imaging process. In this case, the Gaussian mixture model prior distribution on the vector of estimates will be given by:

$$p(\mathbf{x}) = \sum_{i=1}^{K'} \phi_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad (7)$$

where K' denotes the total number of colors or model components such that $K' = K + K_f$, and K_f denotes the number of new (fake) colors generated during image acquisition. In fact, even the human vision system can perceive extra/fake colors due to the imaging conditions. One example is the white reflection we perceive when looking at black leather items. Another problem that perturbs the Gaussian mixture model is the use of 8 bits per pixel by both imaging devices and computers. This 8-bit representation for each channel results in truncating the pixel values close to the lower-bound and/or upper-bound, *i.e.* values close to 0 or 255 for an 8-bit per pixel image. This indicates that there will always be some incorrect color distributions in real-world images. This suggests that the estimation of K' has to always be heuristic.

It would be useful if we can adopt the model we derived in (7) to extract colors from clothing items. Although unsupervised clustering from Gaussian mixture models of (7) can be learned using Bayes' theorem, it is difficult to estimate the Gaussian mixture model of the colors without knowing the colors' ground-truth and the value of K' . Moreover, these models are usually trapped in local minima [16]. Luckily, it has been shown in [26] that k-means clustering can be used to approximate a Gaussian mixture model. We are going to make use of k-means as part of our multistage method, and the full pipeline is illustrated next.

2.2 Clothing Instance Segmentation

We train a Mask-RCNN model [15] to segment all clothing items from an input image. Let this procedure be denoted as:

$$S = \mathbb{M}(\mathbf{f}), \quad (8)$$

where:

- \mathbf{f} denotes the input image, which is a vector of (triplet) RGB values.
- \mathbb{M} denotes the Mask-RCNN model.
- $S = \{\mathbf{s}_0, \mathbf{s}_1, \dots\}$ is a set of images. Each element in S is a vector that denotes a segmented clothing item. We remove the background from each element in S and we store it as a vector of (triplet) RGB values.

Then we use the trained Mask-RCNN model to segment clothing items at color extraction phase.

2.3 Extracting the Main Colors

After segmenting the clothing items of \mathbf{f} , we extract the main/dominant colors in each of them. Our color extraction still has a few phases. We first use a clustering algorithm to obtain the main RGB color components in the clothing segment. Clustering allows us to reduce the number of colors in the image to a limited number. We denote this procedures as follows:

$$\mathbf{c} = \Psi_k(\mathbf{s}), \quad (9)$$

where:

- Ψ_k denotes the clustering model.
- \mathbf{s} denotes a vector of one clothing item that we obtain via (8); we drop the subscript i from \mathbf{s}_i to simplify notation.
- $\mathbf{c} = [c_0, c_1, \dots, c_k]$ is the resultant cluster centers.

Let N_{c_i} be the number of pixels of color c_i and N_s be the total number of pixels in \mathbf{s} , we estimate the probability of each color as follows:

$$p_i = \frac{N_{c_i}}{N_s}, \quad (10)$$

where $i = 0, 1, \dots, k$. Let $\mathbf{p} = [p_0, p_1, \dots, p_k]$ be a vector that contains the probability values of each color in \mathbf{c} .

Clearly, the number of resultant colors equals to the designated number of clusters, k . We use k-means++ clustering algorithm [2, 25] to obtain the main RGB color components in the clothing segment. k-means++ is highly efficient and able to converge in $O(\log k)$ and almost always attains the optimal results on synthetic datasets [2]. We also use fuzzy c-means clustering [6] as an additional comparison method. Fuzzy c-means is much slower than k-means but it is believed to give better results than the k-means algorithm [6].

The main problem in clustering algorithms is the estimation of the value of k , *i.e.* how many colors are there in the clothing item. Choosing a low value will result in incorrect color values if the image has more colors than k . Choosing a high k value will result in more colors than expected, some of which have the same color but differ in tint/shade. Hence, our approach is to choose a high k value, and then merge the colors according to the hue value and probability of each color. This is illustrated next.

2.4 Merging Pure Colors that Have Different Tints/shades

Our next step is to determine whether or not to merge the resultant colors \mathbf{c} in (9). Towards this end, we use a 1D clustering approach that relies on the variations between different color hues. This allows us to merge the colors based on hues that are similar, if any. We denote the procedure as follows:

$$G = \Phi(\mathbf{h}), \quad (11)$$

where:

- Φ denotes a 1D clustering method.
- $\mathbf{h} = [h_0, h_1, \dots, h_k]$ is a vector containing the hue components calculated for each color value of \mathbf{c} .
- G is a set containing subsets $\{G_0, G_1, \dots\}$. Each subset has labels grouped according to the hue values such that: $g_i \in \{0, 1, \dots, k\}$ and $G_i \cap G_j = \emptyset$.

To implement Φ , we try two different methods:

- (1) Mean-Shift [30]. The Mean-Sift method is feature-space analysis technique that can be used for locating the maxima of a density function. It has found applications in cluster analysis in computer vision and image processing [11, 24].
- (2) We also propose a novel 1D clustering algorithm based on differentiating the hue values, and then finding the extreme points according to the maxima of the derivative. We do this as follows:

$$G = \underset{index}{\operatorname{argmax}} \mathbf{h}', \quad (12)$$

where \mathbf{h}' denotes the derivative of \mathbf{h} .

2.5 Probabilistic Modelling

After clustering and color merging, we use the color probabilities given in (10) to estimate the final color. Our suggestion here is driven by the fact that colors with

higher likelihood should dominate the final color. Accordingly, we propose to extract the colors according to the following probabilistic model:

$$d_j = \mathbb{E}_{i \in G_j} [\mathbf{p} \times \mathbf{c}], \quad (13)$$

where $j = \{0, 1, \dots, |G|\}$, \mathbf{p} is given in (10) and \mathbf{c} in (9). The Mean-Shift method also results in a G set, one can similarly use (13) to estimate the colors. The final colors' probabilities can then be calculated using:

$$\hat{p}_j = \frac{N_{d_j}}{N_s}, \quad (14)$$

where N_{d_j} is the number of pixels of color d_j . Hence, our merging algorithm averages the RGB values according to the category and probability of each hue. As a simple example illustrating this approach, one can imagine the color produced by mixing 10% dark blue with 90% light blue in oil paints.

2.6 Color Names

We use the CIELAB color space to match a query RGB value to a lookup table of color names, *i.e.* color names dictionary. It is believed that CIELAB color is designed to approximate human vision [5, 21]. For a query color value, the best match is the color with the minimal Euclidean distance in CIELAB color space to a color in a dictionary of colors. We build the color name dictionary from "Color : universal language and dictionary of names" [19].

3 Results

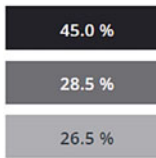
We perform a few experiments to investigate the methods we propose. To reduce clutter, we do not show the color names in most of the figures. We opt instead to provide the probability associated to each extracted color. As there are no color ground-truth, the performance is, unfortunately, subjective. The implementation code can be found in [1].

3.1 The Effect of Number of Clusters

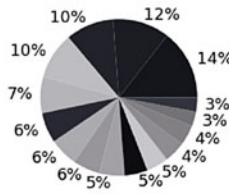
Figure 2 illustrates the effect of only using the k-means compared to our multistage probabilistic approach. Using $k = 20$, one stage k-means extracts different degrees



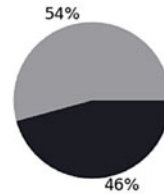
(a) Input image



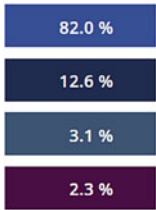
(b) Dress colors via TinEye



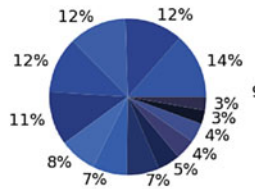
(d) kmeans



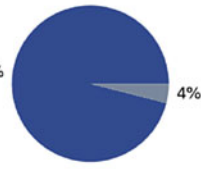
(e) kmeans+Diff



(c) Jacket colors via TinEye



(f) kmeans



(g) kmeans+Diff

Fig. 2 Use of k-means versus our proposed multistage method ‘k-means+Diff’. We use $k = 20$ in all these tests. The dress has three colors; white, black, and gray as shown in (a). Our method is able to extract two colors of which 46% black and 54% gray as shown in (e). Using only k-means gives 16 shades of gray as shown in (d). The jacket has one color (blue), and we are able to extract 96% blue and 4% gray as shown in (g). Using only k-means on the Jacket gives 13 shades of blue as shown in (f). For comparison, we show the results of using the TinEye Color Extraction Tool in (b) and (c). Best viewed in color

of blue from the ‘jacket’ that the lady worn, and different degrees of black and gray from the dress. For the same jacket, our method results 94% blue and 4% gray. For the dress, our method results in 54% light gray and 46% black. In all our models, we remove the color if its probability is less than $1/2k$. We do this to reduce the number of colors based on color probability, *i.e.* as low probability indicates trivial/incorrect color. This is one reason we get lower than k colors when only the k -means is used. Clearly, the multistage clustering we propose helps reduce the colors further as we merge colors of similar hues.

3.2 Comparison with Color Extraction Tools

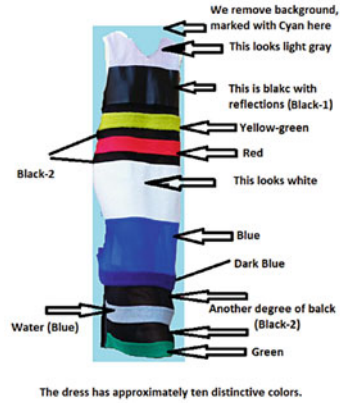
We compare the colors we extract using our proposed method with some of the available commercial tools. For this purpose, we choose Canva (canva.com) and TinEye MultiColor Engine (tineye.com). Canva is one of the commercial design tools that is equipped with a color extractor. TinEye is an image search engine that is based on color features. It must be noted that our color comparisons are subjective because the color ground-truth values do not exist. Figure 3 shows a comparison of our method with that of TinEye. The input image is a picture of a lady wearing a multi colored dress. To highlight these colors for the reader, we manually marked at least 10 colors in the segmented dress, shown in Fig. 3. Results of using TinEye MultiColor engine does not produce white, dark blue, and other degrees of black. Our proposed methods outperform the TinEye MultiColor Engine as the latter and was not able to extract white color and different degrees of black. Moreover, our method extracts more representative colors than that of TinEye. In Fig. 4, we compare our method with Canva tool. Again, our method is able to extract better representative colors. The upper part of Fig. 4 shows that Canva tool did not extract one of the colors in the t-shirt (the ‘Jazzberry jam’ color) and also produce some false colors (Silver and Dark Slate).

3.3 Color Distributions of Fashion Data

Investigating the color distribution of fashion is important. It can be used, among other things, to explore fashion trends. We present in Fig. 5 color distributions of selected items of ModaNet. Although each item may have more than one color, we present the distributions of the colors with the highest probability, these denote the dominant colors. Using only colors with the highest probability simplifies the graphs. We color each point, mimicking one item, according to the extracted color. These graphs provide a nice visual representation about fashion items.



(a) Input image



ColorMarked.png

(b) Segmented dress



(c) Dress colors via TinEye

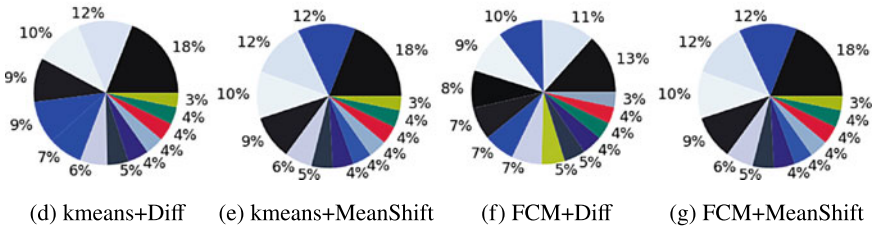


Fig. 3 Comparison of color extraction methods. We manually marked at least 10 colors in the segmented dress shown in (b), Results of a commercial color extraction tool are shown in (c). Variants of our proposed method are shown in (d), (e), (f) and (g); FCM denotes using Fuzzy-C-Means. Our proposed methods outperformed the Multicolor Engine (tineye.com), as the latter was not able to extract the white color and different degrees of black. Best viewed in color



Fig. 4 Comparison of our proposed method with Canva color extraction tool. In each row, colors extracted using our method (lower-right pie-chart) and Canva commercial package (top-right array-chart; www.canva.com). We extract colors from one clothing item segmented out via Mask-RCNN model. To reduce clutter, we did not include the color names, although we can obtain them via the *ColorNames* class. For example, colors of the t-shirt image are: ‘Dark cornflower blue’, ‘Rufous’, ‘Blue sapphire (Maximum Blue Green)’, and ‘Jazzberry jam’. Best viewed in color

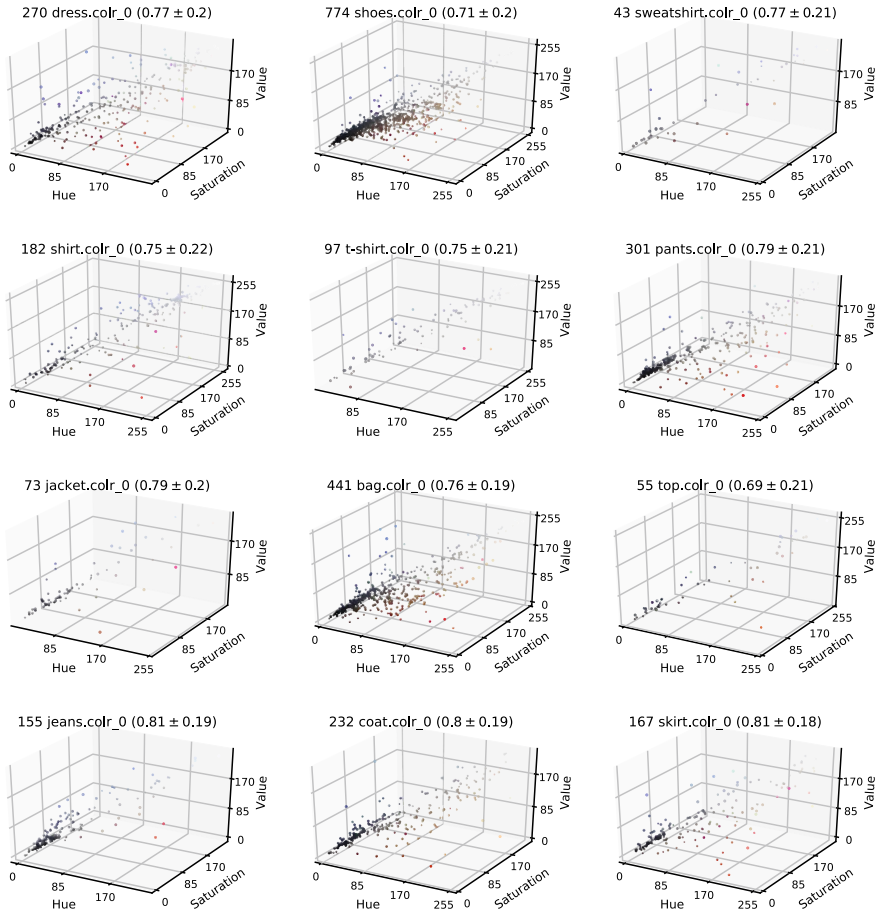


Fig. 5 Estimated color distributions in HSV space. Distributions generated for each item from Clothing Co-Parsing dataset. Titles above each sub-figure are: number of clothing item, item name, *colr_0* is the dominant color (highest probability), followed by mean \pm standard-deviation of probabilities. Best viewed in color

3.4 Fashion Color Trend

We make use of the methods we propose to obtain fashion color trend from a set of images. We extract the color trend from Chanel’s 2020 Spring-Summer season. The images we use are from <https://bit.ly/2WQzKwp>. We use Mask-RCNN that we trained with ModaNet dataset to segment clothing items. Each pie-chart denotes one segmented item. We present the results in Figs. 6, 8, and 7.

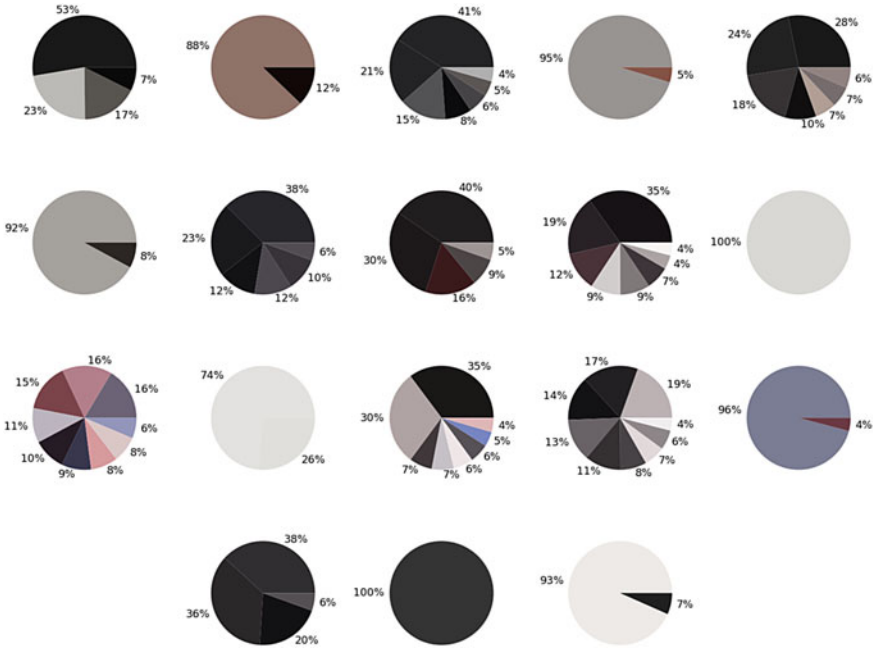


Fig. 6 (Dress) Fashion color trend we extracted from Chanel’s 2020 Spring-Summer season. We use Mask-RCNN that we trained with ModaNet dataset to segment Dresses. Each pie-chart denotes one segmented item. Original model images can be reached via <https://bit.ly/2WQzKwp>. Best viewed in color

4 Discussion and Conclusion

4.1 Gaussian Mixture Model Versus K-Means

Although the k-means method has long been used in color extraction, we wanted to find a link between k-means and the color model we present in (6). It seems that a recent study [26] has found that link. The authors showed that k-means (also known as Lloyd’s algorithm) can be obtained as a special case when truncated variational expectation maximization approximations are applied to Gaussian mixture models with isotropic Gaussians. In fact, it is well-known that k-means can be obtained as a limit case of expectation maximization for Gaussian mixture models when $\sigma^2 \rightarrow 0$ [26]. But according to our color model in (1), $\sigma^2 \rightarrow 0$ indicates that the clothing item is imaged in an idealistic conditions; and that’s intractable in reality. Nevertheless, the work of [26] gives some legitimacy and justification for using the k-means as an approximation of Gaussian mixture models.

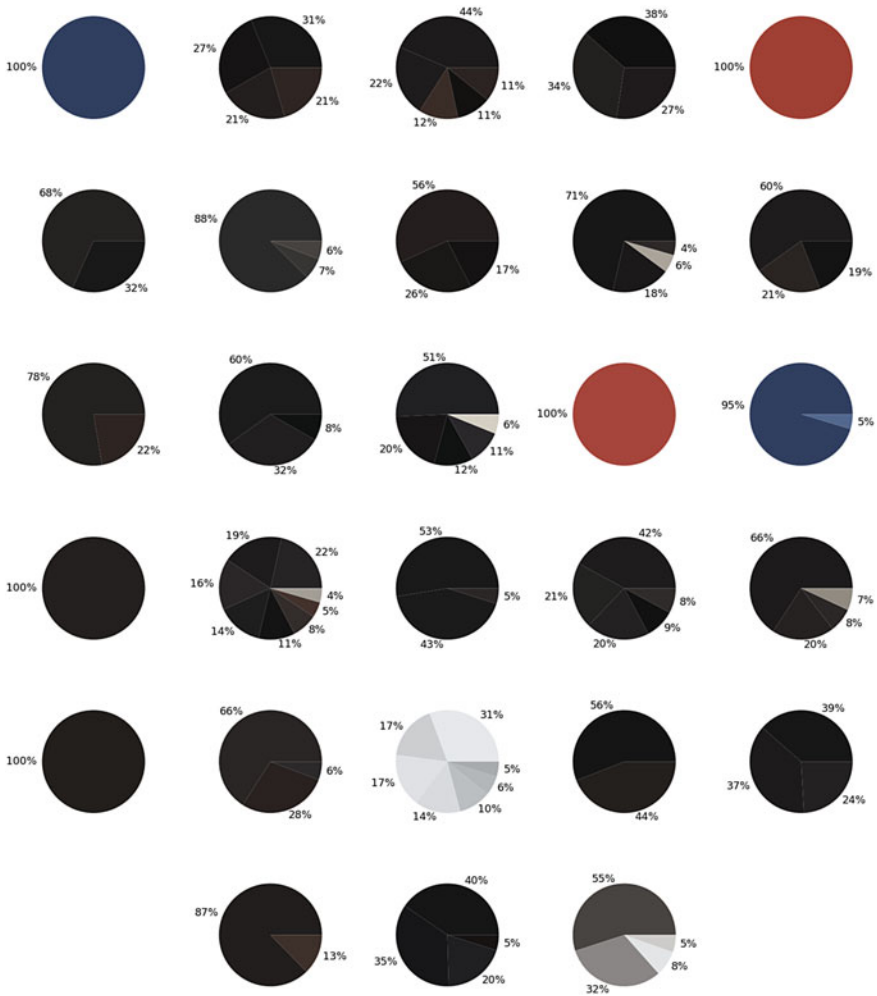


Fig. 7 (Pants) Color fashion trend that we extracted from Chanel’s 2020 Spring-Summer season. We use Mask-RCNN that we trained with ModaNet dataset to segment Pants. Each pie-chart denotes one segmented item. Original model images can be reached via <https://bit.ly/2WQzKwp>. Best viewed in color

4.2 Probabilistic Color Model

One of the biggest problems in color extraction from digital images is dealing with the many colors that a color extractor may reveal. Each clothing item has a set of colors that are, at least, distinctive to the human vision system. And even if we exclude clothes of complex colors and a fractal pattern shapes, the problem remains. One can think of the clustering approach as finding the colors’ outcome by averaging

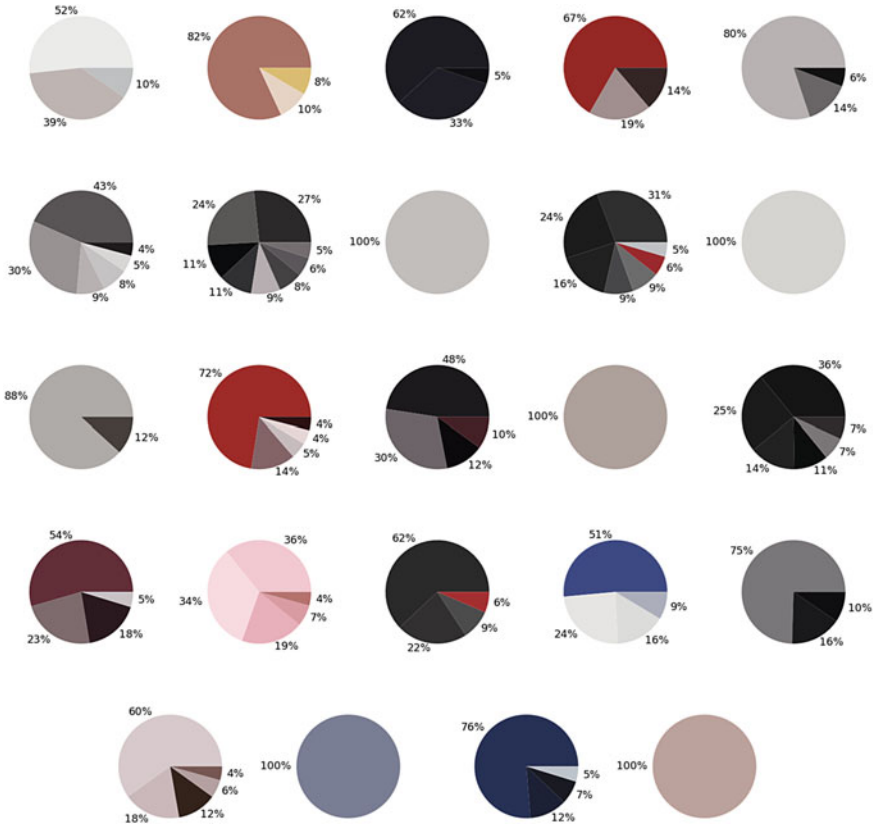


Fig. 8 (Outer) Fashion color trend that we extracted from Chanel’s 2020 Spring-Summer season. We use Mask-RCNN that we trained with ModaNet dataset to segment Outer (coat/jacket/suit/blazers/cardigan/sweater/Jumpsuits/ Rompers/vest). Each pie-chart denotes one segmented item. Original model images can be reached via <https://bit.ly/2WQzKwp>. Best viewed in color

the neighboring color values. However, using a low k value may result in averaging different colors of the clothing item. This may lead to losing the pure colors, if any. Setting k to a high value may result in more colors, such that some colors have similar hues but different shades and/or tints. This variation is expected due to the imaging conditions, amount of light and shadows. Therefore, using a high k value is a better option because it can generate more pure colors. Then, reducing the number of pure colors based on the hue values and associated probability is one way that we find successful to generate more representative colors. The probabilistic model, $\mathbb{E}[\mathbf{p} \times \mathbf{c}]$, we propose in (13) not only performed well in merging colors of similar hues, but has also a natural intuition. This can be justified by the fact that colors appear in a probabilistic manner and can therefore be extracted in the same way.

4.3 Making Use of Color Extraction in Fashion

It is possible to make use of the extracted colors to build color distributions and trends. A color distribution provides a global view of the colors used in a fashion collection. This can be used across different fashion collections in order to have an idea about the colors in each collection. We can see for example how the shoes and bag distributions have a similar shape, which might reflect the color matching between the two groups.

Color value extraction can be highly beneficial if augmented with other predictions obtained via deep learning. Such augmentation can be used to retrieve clothing items from online stores in the event that the items do not have color tags; Or, when items contain many colors in certain percentages. For example, some customers, or fashion designers, may be interested in searching online stores for an item that has the following colors: 40% navy blue, 30% gold yellow, 20% baby pink, and 10% Baby blue eyes. Such palette can be extracted from an item they have seen; similar to the street2shop use-case paradigm [20]. In addition, this probabilistic color palette can be used to build fashion matching models that can be used as part of recommender systems. Clearly, colors extracted from clothes can find numerous applications in the fashion industry. They can be used to help designers in their work, to retrieve clothing items alongside other attributes predicted by AI models, be part of personal shopping and styling apps, and as a vital component of recommender systems.

4.4 Color Perception and Evaluation

Color is usually stored as triplet values in images identified by the red, green and blue channels. However, obtaining the churn of these triplet values via deep learning is a difficult task, although it may seem simple. This is because the extracted colors should conform to the color perception in human vision. Therefore, regression will be better than prediction models. Furthermore, colors should be provided as palette or theme values and not as a few number of tags; which is not yet the case in several fashion datasets. The major problem is that (1) a model that extracts the exact color values is not available; (2) the colors ground-truth is also not available; and (3) if we build a model that extracts the color values, we do not have the colors ground-truth to verify it. This led us to use a subjective measure to judge colors extracted from clothing items. It must be noted that the use of subjective scales may be problematic due to the wide range of millions of colors and differences of opinion.

4.5 Future Prospects

There are many ways to improve this work. For example, if we know the material or fabric type of the piece of clothing, then we can extract colors based this trait. We can do this using a parametric model that takes into account the color that we want to recover according to a parameter denoting the trait. Returning to the Gaussian model mentioned earlier, we can rewrite (3) as follows:

$$\mathcal{N}(x|c_i, \sigma, \gamma) = \frac{1}{|\gamma\sigma|\sqrt{2\pi}} \exp\left(\frac{-(x-c_i)^2}{2\gamma\sigma}\right), \quad (15)$$

where γ is a parameter that one can use to modulate the material type. This is based on the perception that the materials used in the clothes affect the amount of light the camera sensor receives. For example, when leather is the material in use, there will be quite a few reflections camouflaging the real color. Hence, color extraction methods would benefit from γ to either estimate the correct number of colors in the item, or quantify the degree of glittering or shininess. This case would also be interesting if a customer wants to search and retrieve an item with these glittering and shininess traits.

We previously indicated that evaluating color extraction methods is an intractable task unless the data set and its ground-truth are derived from a large number of colors. We aim in our next work to create a large dataset characterised by a large number of colors. Color values are derived from several distributions and are then manipulated to generate true colors. This is done by applying filters that simulate the imaging process. These filters can be generated using image processing tools; or empirically via acquiring printed versions of computer generated source color images. The print quality, imaging device, and many other conditions (*i.e.* indoors or outdoors imaging locations) will affect the acquired images and the computed inverse filter(s). This way we will be able to generate the source, computer generated, and (degraded) ground-truth colors. The dataset can then be used for comprehensive evaluation of our probabilistic color extraction method, as well as other color extraction methods.

Acknowledgements This research was conducted with the financial support of European Union's Horizon 2020 programme under the Marie Skłodowska-Curie Grant Agreement No. 801522 at the ADAPT SFI Research Centre, Trinity College Dublin. The ADAPT SFI Centre for Digital Content Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant No. 13/RC/2106_P2.

References

1. Al-Rawi M (2020) <https://github.com/morawi/FashionColor-0>
2. Arthur D, Vassilvitskii S (2007) K-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, SODA '07, p.

- 1027–1035. Society for Industrial and Applied Mathematics, USA (2007)
3. Bell SM, Liu Y, Alsheikh S, Tang Y, Pizzi E, Henning M, Singh KK, Parkhi OM, Borisyuk F (2020) Groknet: Unified computer vision model trunk and embeddings for commerce
 4. Berg T, Bell S, Paluri M, Chtcherbatchenko A, Chen H, Ge F, Yin B (2020) Powered by ai: Advancing product understanding and building new shopping experiences. <https://tinyurl.com/yyomybc9>
 5. Best J (2017) Colour Design Theories and Applications. A volume in Woodhead Publishing Series in Textiles. Elsevier Ltd. : Woodhead Publishing, Duxford, United Kingdom
 6. Bezdek JC, Ehrlich R, Full W (1984) Fcm: The fuzzy c-means clustering algorithm. *Comput. Geosci.* 10(2):191–203
 7. Biron B (2019) Amazon launched a new personal-styling service that works a lot like stitch fix. <https://tinyurl.com/yy86ajhj>
 8. Capitaine HL, Frèlicot C (2011) A fast fuzzy c-means algorithm for color image segmentation. In: Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-11), pp 1074–1081. Atlantis Press, Aix-les-Bains, France (2011). <https://doi.org/10.2991/eusflat.2011.9>
 9. Cheng WH, Song S, Chen CY, Hidayati SC, Liu J (2020) Fashion meets computer vision: a survey
 10. Co SF (2020) Personal styling for everybody. <https://www.stitchfix.com/>
 11. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. *IEEE Trans Pattern Anal Mach Intell* 24(5):603–619
 12. Delon J, Desolneux A, Lisani JL, Petro AB (2005) Automatic color palette. In: IEEE International conference on image processing 2005, vol 2, pp II–706. IEEE, Genova, Italy (2005)
 13. Delon J, Desolneux A, Lisani JL, Petro AB (2007) Automatic color palette. *Inverse Prob Imaging* 1(2):265–287
 14. EasySize (2020) <https://www.easysize.me/>
 15. He K, Gkioxari G, Dollár P, Girshick, RB (2017) Mask r-cnn. In: Proceedings of the 2017 IEEE international conference on computer vision (ICCV), pp 2980–2988. IEEE, Italy (2017)
 16. Jin C, Zhang Y, Balakrishnan S, Wainwright MJ, Jordan MI (2016) Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In: Proceedings of the 30th international conference on neural information processing systems, NIPS'16, p. 4123–4131. Curran Associates Inc., Red Hook, NY, USA (2016)
 17. Judd DB, Wyszecki G (1975) Color in business, science and industry. Wiley Series in Pure and Applied Optics (third ed.). New York: Wiley-Interscience, New York, NY (1975)
 18. Kang JM, Hwang Y (2018) Hierarchical palette extraction based on local distinctiveness and cluster validation for image recoloring. In: Proceedings of 2018 ICIP, pp 2252–2256
 19. Kelly, KL, Judd DB (1976) Color: universal language and dictionary of names. NBS special publication - 440. Department. of Commerce, National Bureau of Standards, Washington, DC: UD
 20. Kiapour M, Han X, Lazebnik S, Berg A, Berg T (2015) Where to buy it: Matching street clothing photos in online shops. In: 2015 IEEE international conference on computer vision (ICCV), pp 3343–3351. IEEE Computer Society, Santiago, Chile
 21. Kremers J, Baraas RC, Marshall NJ (2016) Human Color Vision. Springer Series in Vision Research, vol 5, 1 edn. Springer International Publishing, Cham, Switzerland (2016)
 22. Lasserre J, Bracher C, Vollgraf R (2018) Street2fashion2shop: Enabling visual search in fashion e-commerce using studio images. In: Marsico MD, GS di Baja, ALN Fred (eds) Pattern recognition applications and methods - 7th international conference, ICPRAM 2018, January 16–18, 2018, Revised Selected Papers, Lecture Notes in Computer Science, vol. 11351, pp 3–26. Springer, Madeira, Portugal (2018). https://doi.org/10.1007/978-3-030-05499-1_1
 23. Lin S, Hanrahan P (2013) Modeling how people extract color themes from images. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI '13, p. 3101–3110. Association for Computing Machinery, New York, NY, USA (2013). <https://doi.org/10.1145/2470654.2466424>

24. Liu Y, Li SZ, Wu W, Huang R (2013) Dynamics of a mean-shift-like algorithm and its applications on clustering. *Inf Process Lett* 113(1):8–16
25. Lloyd SP (1982) Least squares quantization in pcm. *IEEE Trans Inf Theory* 28:129–136
26. Lücke, J, Forster D (2019) k-means as a variational em approximation of gaussian mixture models. *Pattern Recogn Lett* 125, 349–356
27. Manfredi M, Grana C, Calderara S, Cucchiara R (2013) A complete system for garment segmentation and color classification. *Mach Vis Appl* 25:955–969
28. SizerMe (2020) <https://sizer.me/>
29. Yazici VO, van de Weijer J, Ramisa A (2018) Color naming for multi-color fashion items. In: Rocha Á, Adeli H, Reis LP, Costanzo S (eds) *Trends and advances in information systems and technologies*. Springer International Publishing, Cham, pp 64–73
30. Cheng Yizong (1995) Mean shift, mode seeking, and clustering. *IEEE Trans Pattern Anal Mach Intell* 17(8):790–799
31. Yu L, Zhang L, van de Weijer J, Khan FS, Cheng Y, Párraga CA (2017) Beyond eleven color names for image understanding. *Mach Vis Appl* 29:361–373
32. ZeeKit (2020) <https://zeekit.me/>
33. Zhang Q, Xiao C, Sun H, Tang F (2017) Palette-based image recoloring using color decomposition optimization. *IEEE Trans Image Process* 26:1952–1964