Defining a deep neural network ensemble for identifying fabric colors

Alessia Amelio[1], Gianluca Bonifazi[2], Enrico Corradini[2], Simone Di Saverio[3], Michele Marchetti[2], Domenico Ursino[2*], and Luca Virgili[2]

[1] INGEO, University "G. D'Annunzio" of Chieti-Pescara
[2] DII, Polytechnic University of Marche
[3] Imola Informatica
[*] Contact Author
alessia.amelio@unich.it; g.bonifazi@univpm.it; e.corradini@pm.univpm.it; sdisaverio@imolainformatica.it; m.marchetti@pm.univpm.it; d.ursino@univpm.it; luca.virgili@univpm.it

Abstract

Colors characterize each object around us. For this reason, the study of colors has played a key role in Artificial Intelligence (think, for instance, of image classification, object recognition and segmentation). However, there are some topics about colors still little explored. One of them concerns fabric colors. This is a particular topic since fabrics have some characteristics, such as specific textures, that are not found in other contexts. In this paper, we want to propose a new Convolutional Neural Network (CNN) based model for identifying fabric colors. After introducing this model, we consider three different versions of it and create an ensemble of the corresponding CNNs to get better results. Finally, through a series of experiments, we show that our ensemble is able to improve the state-of-the-art on the identification of fabric colors.

Keywords: Color Classification; Ensemble Learning; Identification of Fabric Colors; Classification of Fabric Colors; Convolutional Neural Networks

1 Introduction

Visual information is one of the most important mechanisms by which people communicate (think, for instance, of documents, images, videos, movies, etc.). Color is an essential component of visual communication. It is also a relevant topic in several research fields, for example in physics to model light and reflectance, in physiology to model perception, in biology to model the visual system, in art to model beauty. As a result, the study of colors has fascinated, and continues to fascinate, researchers from a wide variety of fields [21, 31].

One of the general problems with colors concerns the identification of the color of objects and, more generally, of any entity. This issue is critical because it allows, for example, the characterization of the content of objects, the understanding of emotional and psychological states, and the discovery of recurrent patterns and anomalous conditions.

One area in which color recognition is essential is textile industry. In this scenario, color is one of the most important information to be transmitted between the manufacturer and the customer.

Furthermore, in the same industry, it is one of the most important characteristics that can be leveraged in marketing and R&D, as well as one of the most critical features that designers employ to succeed. The attraction that the color of a fabric can exert in a customer's final decision has also been studied scientifically (see, for instance, [27]).

In recent years, several authors have investigated the issue of color identification in several fields including the textile industry. This has led to various methods for recognizing color components, patterns and shades, and for automatically detecting the layout of color yarns from images. These methods use several approaches such as fuzzy C-Means [43, 18] and X-means [7], rule-based techniques [1], Support Vector Machine [28, 45], Random Forest [5] and deep neural networks [25], to reach their objectives. Other works have focused on textile whiteness estimation [33]. Others have investigated color prediction and recognition using deep neural networks [14, 47, 16, 38, 2] and XGBoost [48]. Finally, several works on color recognition match the color of interest with a reference palette to detect the tolerance with respect to fabric standards [36, 41].

In the past, deep learning based approaches have been used for color identification in different contexts, including color recognition of vehicles and traffic lights [47, 16], flowers [2], stool medical images [38], etc. In all of these cases, they performed better than other machine learning approaches [47]. Despite this, the use of deep learning for color recognition in the context of textile is still poorly investigated. Moreover, the proposed approaches have some limitations in extracting colors within recurrent patterns, shades and layout variations, which are very important features for fabric analysis.

In this paper, we want to make a contribution in this setting by proposing a new deep learning based approach for fabric color identification. To overcome the limitations described above, we propose a new architecture based on an ensemble of Convolutional Neural Networks (hereafter, CNNs), whose input is in the color difference domain. This is obtained by considering the difference between the original input image and a set of reference color images. Representing the input in the color difference domain allows color variations, shades and patterns to be easily captured. This can make easier for our network to learn features for color recognition in fabric images. Furthermore, adopting an ensemble strategy provides greater robustness than a single CNN, which, in turn, results in greater accuracy in returned results.

We focus on one particular aspect of color recognition in the textile industry, namely color identification of fabric images. This issue is interesting to address because it can help reduce inefficiency, wasted time and subjectivity, which can ultimately lead to higher quality products that better meet customer needs. We performed experiments to evaluate our approach on a well-known dataset of fabric images and compared our results with other state-of-the-art deep neural networks for color identification. The results obtained show that a CNN ensemble in the color difference domain not only outperforms single CNNs but also obtains better results than those returned by well-known deep learning architectures widely used for pattern recognition and color identification.

The outline of this paper is as follows: Section 2 reviews the related literature, with a particular focus on the analysis of colors in the textile scenario. Section 3 presents our approach and, in particular, its two phases involving mapping the color space in the color difference space and defining an ensemble of CNNs. Section 4 illustrates the experiments we performed to test and evaluate our approach. Finally, in Section 5, we draw our conclusions and look at possible future developments of our research efforts.

2 Related Literature

Color identification is a challenging image processing task in the context of Computer Vision. In fact, color represents a fundamental feature in the study of digital images and is the starting point for many object recognition and tracking techniques [23, 34]. For this reason, in the past literature, many color recognition techniques have been proposed. They can be divided into three distinct threads, which adopt handcrafted features, machine learning and deep learning, respectively, to reach their objective.

Handcrafted feature based approaches are the most traditional ones because they are easy to implement. They generally have a high execution speed against a lower accuracy, due to a poor generalization capability. One of the most critical features of digital color images is the RGB value of pixels [42, 1, 22, 29]. In this context, the authors of [1] present a rule-based approach to recognize different shades of basic colors of fabric images. They extract the RGB color features to obtain the mean and standard deviation of red, green and blue shades. Then, thanks to these features, they define the rules for color classification considering ten shades for each color. The main drawback of this approach is in the definition of rules, because it is difficult to find effective rules capable of capturing the different color shades. Sometimes, the recognition of colors by means of single features may not provide satisfactory results. For this reason, the authors of [6] propose an approach that combines multiple features. In particular, it combines transformed color histogram, hue histogram, normalized RGB histogram and color moment as feature patches to recover information from each feature.

As for machine learning based approaches, several studies have adopted the Gaussian Mixture Model (GMM, for short) to obtain an unsupervised classifier of color regions [15, 39, 20]. In particular, the authors of [15] present an approach for finding color regions in digital images. It consists of two steps. In the first one, it estimates GMM parameters by applying Expectation Maximization on a reference image that includes the regions of interest. For this purpose, it considers two chrominance features, i.e., Cb and Cr channels in the YCbCr color space model. The second step searches and segments the color regions in the input images using GMM parameters extracted from the reference images. The authors of [48] use the XGBoost classification model to classify the color of solid wood flooring. First they analyze all features; then, they remove those with low variance in order to keep only the most essential information and improve classification speed and accuracy. The two most widely used machine learning algorithms for color classification and clustering are Support Vector Machine and X-means [36, 28, 35, 8, 3]. In [35], the authors present an approach that uses X-means to perform color matching in textiles, whose ultimate goal is measuring quality and increasing efficiency. They show that X-means provides better results than the standard Euclidean distance method for color detection. Recent studies use an evolution of X-means, called fuzzy C-Means. In this case, data points can belong to more than one cluster with a given probability. Thanks to this choice, this approach obtains comparatively better results for overlapped datasets [43, 42, 18].

Deep learning based approaches have been used recently in several domains, such as health care and face and object recognition [38, 9, 16]. Many of them focus on vehicle color recognition [47, 46, 30, 11]. To this end, they train deep learning models capable of recognizing vehicle colors in photos taken in a road environment, despite the presence of adverse conditions, such as the existence of shadows or reflections. In [30], the authors propose a method for vehicle color recognition using Convolutional Neural Networks. They show that this method is robust and accurate in handling the variation of lighting and environment, and outperforms approaches using only traditional handcrafted feature descriptors.

Color recognition approaches are also slowly gaining popularity in the textile and fashion industry, where they are used to accurately identify fabric colors [20, 25]. In [20], the authors focus on fashion product portraits and propose an approach that autonomously finds the garment region in an image and, then, retrieves information on colors and patterns of items. In [25], the authors propose a regression model for color retrieval in fashion garments. Their approach focuses mainly on objects belonging to a different category of garments. To this end, it modulates the importance of each pixel by adopting the so-called color name-attention technique, which selects only those pixels sufficiently close to the main color. This way of proceeding guarantees it a good performance. However, this decreases when the approach has to deal with fabrics without garments, because the color name-attention technique cannot be employed in that case.

3 Description of the proposed approach

We can refer to a color space as a specific organization of colors. To define it, we need to use a color model, which is an abstract mathematical model describing how colors can be represented as tuples of numbers. Examples of color models are RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), CMYK (Cyan, Magenta, Yellow, Key). Currently, the most widely used color space is RGB, which describes the light emitted by a given color through a triplet of numbers representing the combination of three basic colors, namely red, green and blue. Although this model is sufficient to represent digital images, it does not allow the extraction of handcrafted features sufficient to guarantee high performance in the color classification task [42, 1, 6]. For this reason, our approach starts from the RGB model and then maps the images to another domain that represents color difference. This adds more information about the input images and provides additional perspectives that could be leveraged for color recognition.

3.1 Mapping the color space into the difference domain

To the best of our knowledge, the deep learning approaches for color identification proposed in the past literature do not consider textural patterns (which are heavily present in clothes images), since they can hamper the learning of the underlying neural network. This way of proceeding derives at least in part from the color representation used for images [42, 1, 6], which does not provide sufficient information for color classification. To overcome this limitation, we introduce a new color space called difference space. As we will see later, it overcomes the limitations of the classical color spaces in the management of images and greatly simplifies the color recognition issue.

The reasoning underlying the difference space consists in computing the distances between the colors of an input image and a set of reference colors. First, we need to choose a set of p reference colors and determine the corresponding RGB encoding. Then, we compute the differences between the colors of the input image and the reference colors and obtain p images represented using the RGB encoding. After that, we concatenate all the p images thus obtained to create the corresponding input to the deep learning model. This input has the same width and height as the initial images, but it has a third deeper dimension, of size $q = p \times 3$. It is obtained by concatenating the p images, each represented by means of the 3 RGB channels.

Formally speaking, we define a function $\mathcal{F}\colon \mathbb{R}^{m \times n \times 3} \to \mathbb{R}^{m \times n \times q}$, where m and n are the width and height of the image, and q is the size of the difference domain. $\mathcal{F}$ is the concatenation of the element-wise distance between the RGB channels of the input image and the corresponding ones of the reference color:

$$\mathcal{F} = Concat_{i=1}^{p}\{(R_{input} - R_{color_i}), (G_{input} - G_{color_i}), (B_{input} - B_{color_i})\} \tag{1}$$

Here, R_{input}, G_{input} and B_{input} are the corresponding Red, Green and Blue channels of the input image, while R_{color_i}, G_{color_i}, and B_{color_i} are the values of the RGB encoding of the i-th reference color. In order to comply with the RGB model, if a subtraction between the input color and the reference one returns less than 0, we set the corresponding result equal to 0.

For the purpose of this paper (i.e., identifying colors from fabric images), we use 12 reference colors, namely Orange, White, Blue, Cyan, Yellow, Magenta, Black, Red, Earth Brown, Green, Emerald Green, and Purple. Actually, it would have been possible to use other sets of colors. We chose these because, in the context of textiles, they represent most of the possible cases [13]. As a consequence, for each image to be classified, we create 12 images, each representing the distance of the image itself from one of the reference colors. This distance is expressed in the RGB model. To give an idea of our way of proceeding, in Figure 1 we show an example of the differences between an input image and the set of reference colors. Finally, we concatenate the 12 images thus obtained on the third axis of the RGB channels to build the input for the tasks of the next phase.

3.2 Ensemble learning for color identification

In the previous section, we have seen how, by switching to the color difference domain, given a fabric image, we obtain a concatenation of 12 images, each of which can be represented using the RGB model. Since this model has 3 channels, it follows that each image can be represented by 36 channels. To process such data we introduce an extension of the traditional CNN that we call CNN$^\Delta$. It consists of an input layer that receives images of any width and height through 36 channels. After this layer, there is a convolutional layer with 3×3 kernels and stride 1, followed by a max pooling layer of size 2×2 and stride 1. Then, there is a convolutional layer with the kernel size identical to the previous one followed by a pooling layer of size 2×2 and stride 1. After them, we have a dense layer with 128 units connected to a final dense layer with a softmax activation function. The output of this network is a vector of 12 elements each representing the classification probability of the corresponding reference color. The total number of parameters in CNN$^\Delta$ is 175,476. The schematic representation of CNN$^\Delta$ is shown in Figure 2. In this case, it is designed for an input image having a width and a height of 100 pixels.

To further improve the performance and generalization of our CNN$^\Delta$ model, we adopt an ensemble learning technique. To this end, we construct a deep learning model consisting of multiple models operating together to predict the class of an input image. In particular, our model consists of an ensemble of three different versions of CNN$^\Delta$, each having a unique set of hyperparameters and training methods, as shown in Table 1. Overall, the ensemble of the three CNN$^\Delta$ models revealed to be the best trade-off between performance and computational load. In the following, we call $\mathcal{E}_{CNN^\Delta}$ this ensemble.

Network	*Optimizer*	*Learning rate* η	*Momentum*	ϵ
CNN$^\Delta$ 1	ADAM	0.001	-	10^{-7}
CNN$^\Delta$ 2	ADAM	0.0001	-	10^{-8}
CNN$^\Delta$ 3	SGD	0.01	Nesterov	-

Table 1: Main characteristics of the three CNN$^\Delta$ models used in $\mathcal{E}_{CNN^\Delta}$

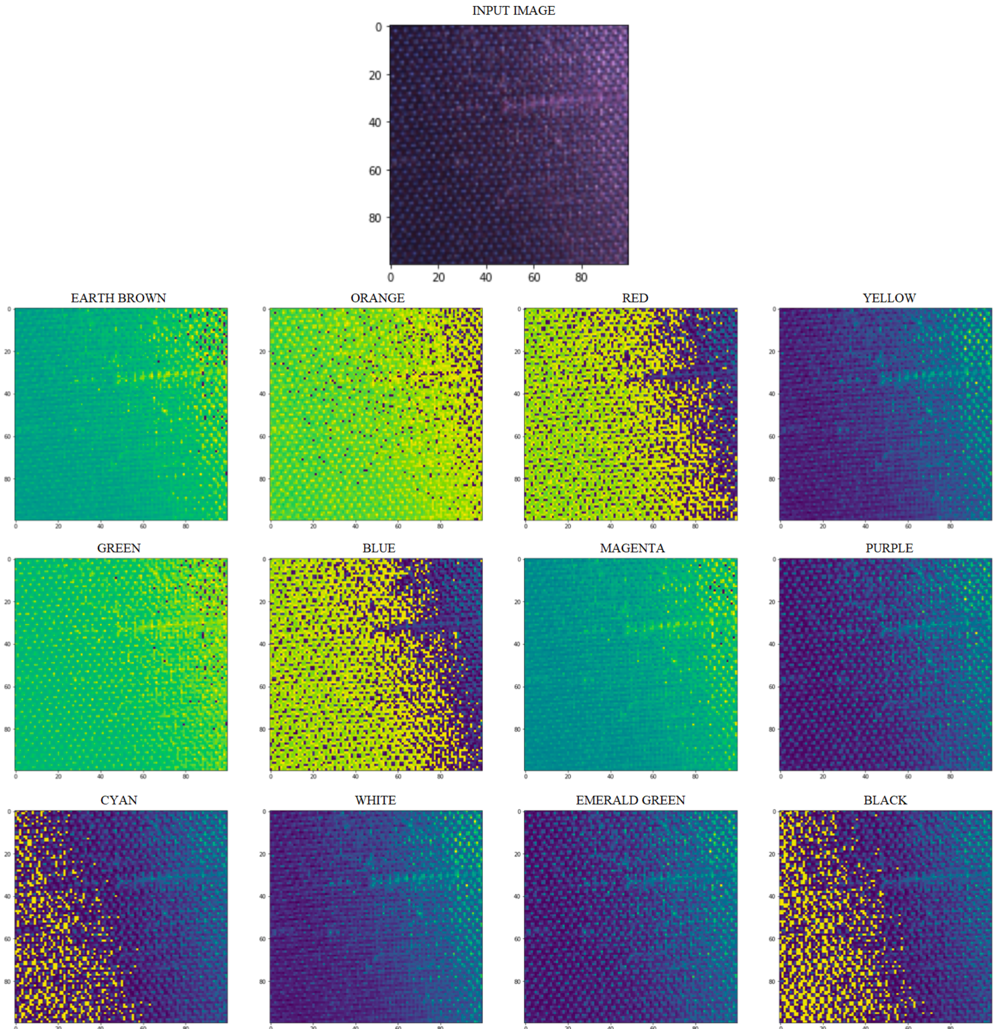

Figure 1: An input image and the differences between it and the set of reference colors

$\mathcal{E}_{CNN^{\Delta}}$ works as follows. During the prediction step, each CNN$^{\Delta}$ provides a classification probability vector for the reference colors. In this way, three probability vectors are obtained. These are processed by a soft-voting function. This function first computes the average of the classification probabilities, then it identifies the highest values thus obtained and, finally, returns the corresponding class as final output. In Figure 3, we report a graphical representation of $\mathcal{E}_{CNN^{\Delta}}$.

4 Experiments

4.1 Testbed

We tested $\mathcal{E}_{CNN^{\Delta}}$ on the Fabrics Dataset [13]. It consists of a benchmark collection of 2,000 different color images related to clothing and textiles. The format of the images is `PNG` and their size is 400×400. They were acquired in clothing stores and labeled with the material composition provided by the manufacturer, as well as with information about the cloth origin (e.g., jacket, sweater, pants).

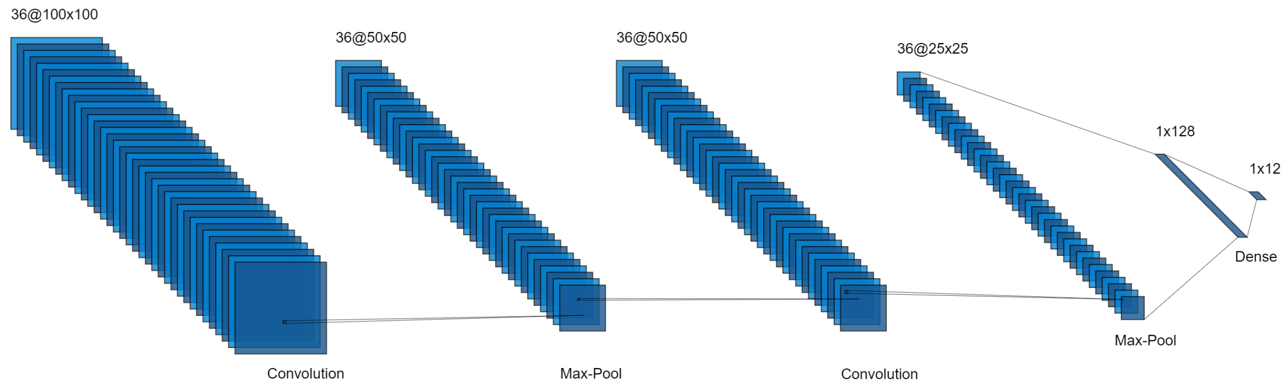

Figure 2: Architecture of CNN$^{\Delta}$ designed for an input image having a width and a height of 100 pixels

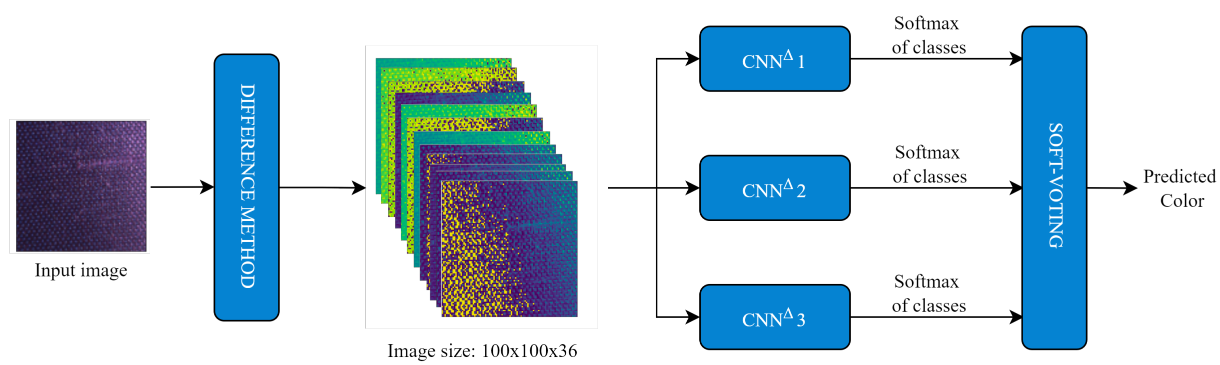

Figure 3: A graphical representation of $\mathcal{E}_{CNN^{\Delta}}$

Each image type was acquired using a photometric stereo sensor under 3 or 4 different illumination conditions, resulting in a total of 7,757 actual images. The dataset is not balanced because it was conceived to represent the real-world distribution of fabrics. Most of the images depict cotton and polyester clothes, while some of them regard silk and linen ones. The garments of many clothes are composed of two or more fabrics (blended fabrics). Figure 4 shows four sample images from the Fabrics Dataset.

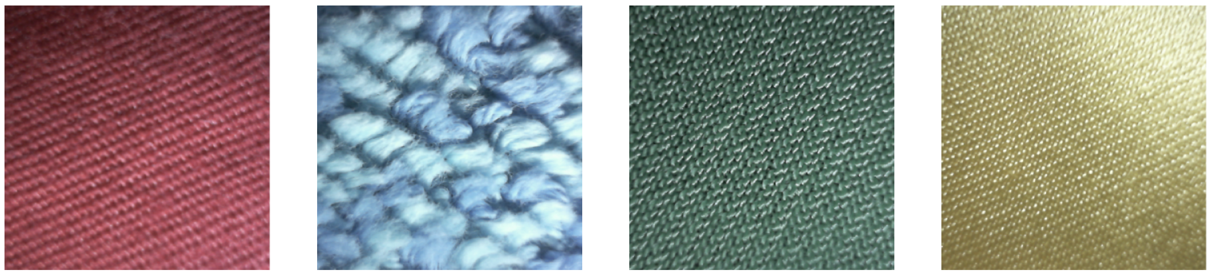

Figure 4: Sample images from the Fabrics Dataset

Since the images in the Fabrics Dataset are not labeled with color, we had to manually specify the ground truth for each of them. For this purpose, we performed a procedure based on a visual inspection of them. Specifically, we showed each image to 11 different people and asked them to label it with one color among the 12 reference ones specified in Section 3.1. We assigned to an image the color most labeled by the 11 people to whom it was shown. We discarded all the images with which no color could be associated.

At the end of this procedure, we kept 4,591 images while discarding 3,166. The distribution of the ground truth colors relative to the images retained is shown in Figure 5.

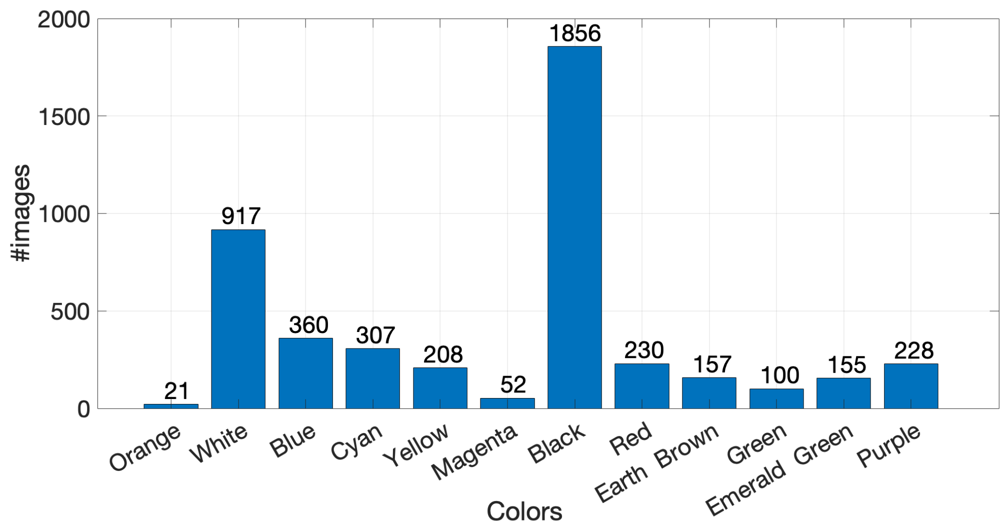

Figure 5: Distribution of the ground truth colors of the images kept in the dataset

Given this collection of images, we applied an undersampling procedure on the most numerous classes and an oversampling procedure on the least numerous ones to achieve a balance of classes. At the end of this procedure, we obtained a new dataset consisting of 1,200 images, in particular 100 images per class. Since images were acquired under different lighting conditions, the color identification task can be made more difficult by the presence of differently illuminated points. For example, this can happen due to shadows and flash glare from reflective fabrics. To address this issue, we first performed a pre-processing step on the 1,200 images for smoothing the differences in illumination caused by shade [44] and flash glare [26, 40]. In the following, we call $\mathcal{D}$ the final dataset thus obtained.

We compared $\mathcal{E}_{CNN^\Delta}$ with different deep learning state-of-the-art architectures. In particular, we considered: *(i)* the deep residual network (i.e., ResNet) [10], *(ii)* the inception network (i.e., Inception-ResNet) [32], and *(iii)* the dense convolutional network (i.e., DenseNet) [12]. We also considered: *(iv)* the Color Deep network [46], which is a reference network in the color recognition literature. Specifically, we selected two types of ResNet, i.e., ResNet50v2 and ResNet152v2, two types of DenseNet, i.e., DenseNet169v2 and DenseNet201v2, and InceptionResNetV2. We chose them because they revealed to achieve the best experimental results. Furthermore, to verify the effectiveness of adopting an ensemble strategy, we compared $\mathcal{E}_{CNN^\Delta}$ with a single CNN$^\Delta$. For all the networks involved in the comparison task, the input consisted of 36 channels representing the difference between the image to be classified and the 12 reference colors (see Section 3.1). Since Color Deep could not simultaneously take in input the 36 channels corresponding to the 12 differences, we gave to this network the 12 images corresponding to the differences in cascade, and then aggregated the corresponding results.

To avoid overfitting in the classification of colors, we applied a data augmentation procedure in the images of $\mathcal{D}$. This activity consists of two tasks, namely: *(i)* random flip augmentation, which flips the image horizontally and vertically, and *(ii)* random zoom augmentation, which randomly increases the image size by adding new pixels around or interpolating the values of pixels.

We provided the original images of $\mathcal{D}$ in input to the various models to be tested. The identification of colors in them is complex because of the presence of textural patterns.

We also extracted only the color components from the original fabric images and generated images with the only dominant color. We call $\mathcal{D}^-$ the corresponding dataset. By the term "dominant color" we mean the most diffuse color in the image area. To extract it from each image we initially exploited the Modified Median Cut Quantization (MMCQ) algorithm [4] with the goal of clustering the color space of the image. Then, we selected as dominant color the one corresponding to the center of the largest cluster thus identified.

At the end of all these procedures, we had two datasets at disposal, namely *(i)* $\mathcal{D}$, storing the images with the original colors, and *(ii)* $\mathcal{D}^-$, storing the same images as $\mathcal{D}$ but pre-processed to contain only the dominant color.

For the Color Deep network, we also employed a transfer learning procedure. Specifically, we pre-trained the network with the Car Dataset [19], which contains 16,185 images of colored vehicles. After the pre-training task, we fine-tuned the Color Deep network with the fabric images for identifying the corresponding color. To label car images with the ground truth color, we chose the color list proposed in [24] and used the approach for extracting the dominant color described above, followed by an approach for performing color search. The latter considered a palette of 865 reference colors in the CIELAB color space. Each color of the palette had associated a name and a hexadecimal code. For a given color image, we computed the Euclidean distance between the corresponding color and each color of the palette. We chose the color corresponding to the minimum distance as the one of the image.

To evaluate the various deep learning models, we used the 5-fold cross validation on $\mathcal{D}$ and $\mathcal{D}^-$. For each model, we computed the average values of Precision, Recall, F1-score and Accuracy.

To define the optimal hyperparameter values for the various neural networks, we used a random search procedure. Furthermore, in accordance with the results found in [17], we considered low values of the ϵ parameter and adopted the ADAM optimizer for CNNs. In Table 2, we report the parameter setting for the different learning models adopted in our experiments. The last three rows of this table refer to the three CNN$^\Delta$ models that compose $\mathcal{E}_{CNN^\Delta}$ and whose parameters were already partially defined in Table 1.

Deep neural network	*Optimizer*	*Learning rate η*	*Decay*	*Momentum*	ϵ
Color Deep	SGD	0.001	10^{-7}	0.9	-
ResNet50v2, ResNet152v2	SGD	0.1	-	0.001	-
DenseNet169v2, DenseNet201v2	SGD	0.1	-	0.001	-
InceptionResNetV2	SGD	0.01	-	0.001	-
CNN$^\Delta$	ADAM	0.001	-	-	10^{-7}
CNN$^\Delta$ 1 of $\mathcal{E}_{CNN^\Delta}$	ADAM	0.001	-	-	10^{-7}
CNN$^\Delta$ 2 of $\mathcal{E}_{CNN^\Delta}$	ADAM	0.0001	-	-	10^{-8}
CNN$^\Delta$ 3 of $\mathcal{E}_{CNN^\Delta}$	SGD	0.01	-	Nesterov	-

Table 2: Parameter setting of the deep neural networks employed in our experiments

To limit overfitting, we monitored the validation loss and stopped the training task when there was no change of validation loss for three consecutive iterations.

Finally, we implemented our experiments by means of the free version of Google Colab, which provided a GPU NVIDIA Tesla K80, 12 GB RAM, and two Intel Xeon CPUs 2.30GHz. The programming environment consisted of Python 3.7, Tensorflow 2.6, OpenCV 4.2 and Scikit-learn 1.0.

4.2 Results

Table 3 shows the values of Precision, Recall and F1-score for each class of colors obtained by applying $\mathcal{E}_{CNN^\Delta}$ on $\mathcal{D}$. Instead, in Table 4, we report the values of these three parameters averaged over all classes, along with the Accuracy value. Finally, Figure 6 shows the corresponding confusion matrix.

Color class	*Precision*	*Recall*	*F1-score*
Orange	**0.95**	**1.00**	**0.98**
White	**0.90**	**0.95**	**0.90**
Blue	0.74	1.00	0.88
Cyan	0.65	0.65	0.65
Yellow	0.70	0.70	0.70
Magenta	**0.95**	**1.00**	**0.98**
Black	**0.95**	**0.95**	**0.95**
Red	0.85	0.85	0.85
Earth Brown	0.75	0.60	0.67
Green	0.76	0.65	0.70
Emerald Green	0.74	0.85	0.79
Purple	0.54	0.35	0.42

Table 3: Precision, Recall and F1-score for each class of colors obtained by $\mathcal{E}_{CNN^\Delta}$ when applied on $\mathcal{D}$

Average Precision	*Average Recall*	*Average F1-score*	*Accuracy*
0.79	0.80	0.79	0.80

Table 4: Average Precision, Average Recall, Average F1-score and Accuracy obtained by $\mathcal{E}_{CNN^\Delta}$ when applied on $\mathcal{D}$

From the analysis of Tables 3 and 4 and Figure 6, we can see that $\mathcal{E}_{CNN^\Delta}$ obtains very satisfactory results. As far as the single colors are concerned, Orange, White, Magenta and Black (bold marked in Table 3) reach the highest values of the performance measures. In fact, their values fall in the range $[0.90, 1]$ and the number of misclassifications involving them in the confusion matrix is very low. Instead, $\mathcal{E}_{CNN^\Delta}$ has difficulty in classifying Purple. This can be seen in both Table 3 and the confusion matrix of Figure 6. Such a behavior may be due to the texture of some fabric images having lighter purple nuances that make the identification of Purple more difficult. In these cases, the corresponding images were classified with colors, like Blue and Cyan, closely related to Purple in the color palette.

To show the advantage of using an ensemble strategy, in Table 5 we report the results obtained from the three CNN$^\Delta$ models separately on the same dataset considered to evaluate $\mathcal{E}_{CNN^\Delta}$ previously.

In this case, we can see that the first network (bold marked) obtains the best results, with performance values in the range $[0.76, 0.77]$. Comparing these values with the ones of Table 4, we can see that the adoption of the ensemble strategy produces an improvement of 2.60% in Precision, 5.26% in Recall, 3.95% in F1-score and 5.26% in Accuracy.

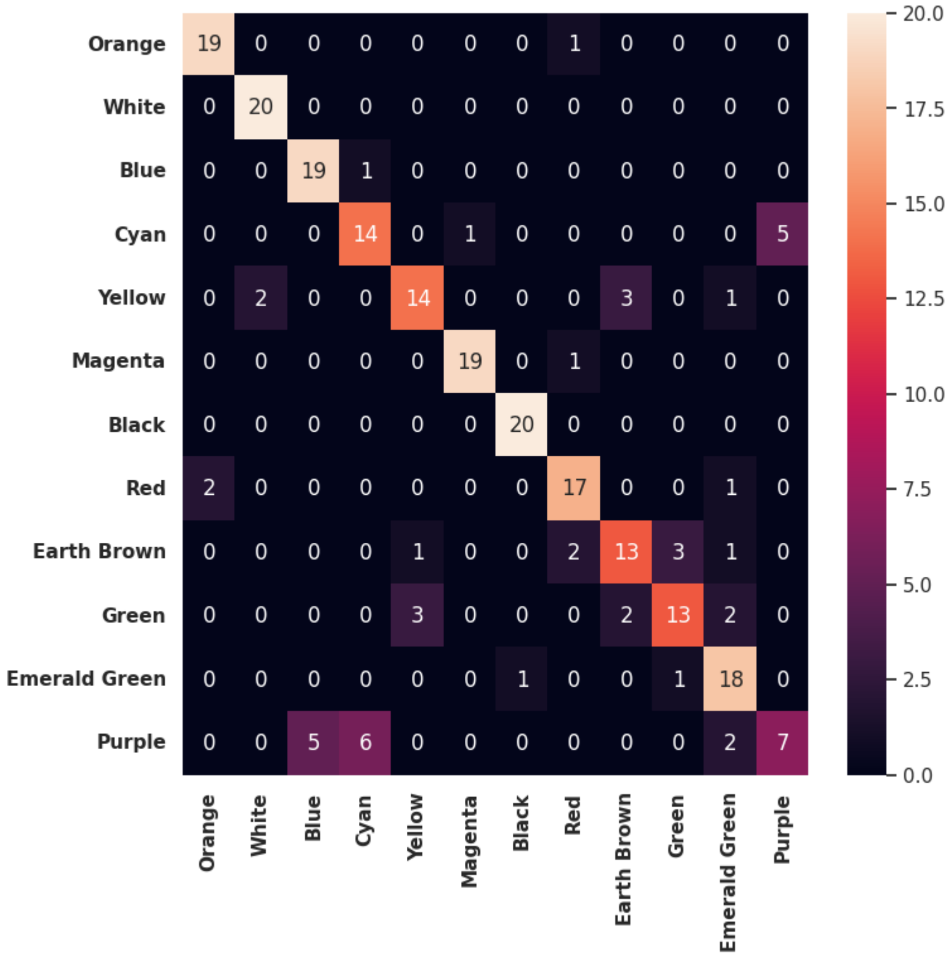

Figure 6: Confusion matrix obtained by $\mathcal{E}_{CNN^{\Delta}}$ when applied on $\mathcal{D}$

4.2.1 Comparisons

Tables 6 and 7 show the results on color identification for our approach and the ones we chose for comparison (see Section 4.1). Specifically, Table 6 refers to the original fabric images (i.e., images of $\mathcal{D}$), while Table 7 refers to the fabric images with the dominant color (i.e., images of $\mathcal{D}^{-}$). In both cases, we used 5-fold cross validation. Furthermore, for Color Deep we considered both the average and the best result.

Table 6 shows that, for the images of $\mathcal{D}$, we can observe that $\mathcal{E}_{CNN^{\Delta}}$ overcomes the other deep learning architectures in all performance measures. Color Deep is able to obtain good performances in its second configuration. However, these are 2-3% lower than those obtained by $\mathcal{E}_{CNN^{\Delta}}$. Another result emerging from Table 6 is that the ensemble strategy obtains better results than the usage of a single CNN$^{\Delta}$. In fact, the value of Precision (resp., Recall, F1-score, Accuracy) obtained by $\mathcal{E}_{CNN^{\Delta}}$ is higher of 2.53% (resp., 5.00%, 3.80%, 5.00%) than the corresponding one obtained by a single CNN$^{\Delta}$.

The examination of Table 7, and its comparison with Table 6, shows that providing input images with dominant color to the models under examination does not lead to improved results. On the contrary, for all the models except $\mathcal{E}_{CNN^{\Delta}}$, a marked decrease is observed. The best results after

CNNs$^{\Delta}$	*Precision*	*Recall*	*F1-score*	*Accuracy*
Network 1	**0.77**	**0.76**	**0.76**	**0.76**
Network 2	0.74	0.75	0.74	0.75
Network 3	0.73	0.73	0.72	0.73

Table 5: Average Precision, Average Recall, Average F1-score and Accuracy obtained by the three CNNs$^{\Delta}$ composing $\mathcal{E}_{CNN^{\Delta}}$ when working individually

Deep neural network	*Precision*	*Recall*	*F1-score*	*Accuracy*
$\mathcal{E}_{CNN^{\Delta}}$	**0.79**	**0.80**	**0.79**	**0.80**
CNN$^{\Delta}$	0.77	0.76	0.76	0.76
ResNet50v2	0.73	0.65	0.69	0.71
ResNet152v2	0.66	0.65	0.62	0.65
InceptionResNetV2	0.72	0.69	0.67	0.69
DenseNet169v2	0.71	0.69	0.67	0.69
DenseNet201v2	0.68	0.66	0.65	0.66
Color Deep (avg.)	0.72	0.72	0.70	0.72
Color Deep (max.)	0.77	0.77	0.76	0.77

Table 6: Results of $\mathcal{E}_{CNN^{\Delta}}$ and the other models chosen for comparison when the images of $\mathcal{D}$ are provided in input

the one of $\mathcal{E}_{CNN^{\Delta}}$ are achieved by the single CNN$^{\Delta}$, whose Precision (resp., Recall, F1-score and Accuracy) value is 11.53% (resp., 15.38%, 14.10% and 16.46%) lower than that obtained by $\mathcal{E}_{CNN^{\Delta}}$. The other models perform even worse and, among them, again the best one is Color Deep in its second version. Once again we can observe that the ensemble strategy obtains better results than the use of a single CNN$^{\Delta}$.

Contrary to what we might have expected, using the images with the dominant color obtains worse results than the ones achieved when employing the original fabric images. This can be due to the flattening of some main color nuances, which increases the difficulty to discriminate the true color by the deep learning models into examination.

Figure 7 (resp., 8, 9) deepens the previous analysis by showing the values of Precision (resp., Recall, F1-score) obtained by applying $\mathcal{E}_{CNN^{\Delta}}$, a single CNN$^{\Delta}$ and Color Deep on the original fabric images. From the examination of this figure we can observe that the Precision of $\mathcal{E}_{CNN^{\Delta}}$ is particularly good for Orange, Magenta, Black, White and Red. Conversely, Color Deep achieves a high Precision for Red, Earth Brown, Green and White. Both models reach a high Precision for White and Red, while both of them obtain a low Precision for Cyan and partly for Yellow. As could be expected from the results reported in Tables 6 and 7, $\mathcal{E}_{CNN^{\Delta}}$ always has a higher Precision than a single CNN$^{\Delta}$.

Instead, Figure 8 shows that the Recall of $\mathcal{E}_{CNN^{\Delta}}$ is very high for Orange, White, Blue, Magenta, Black, and partially for Red, Yellow and Emerald Green. In contrast, the Recall of this model is low for Purple. Regarding Color Deep, we can see that the Recall is very high for Orange, White, Blue, Magenta, Red and Earth Brown and high for Green and Purple. Color Deep reaches the lowest values of Recall for Cyan and Emerald Green. Orange, White, Blue, Magenta, Black and Red are colors with a high Recall for both models. Once again, we can observe that $\mathcal{E}_{CNN^{\Delta}}$ always reaches a better Recall than a single CNN$^{\Delta}$.

Finally, examining Figure 9, we deduce that the F1-score of $\mathcal{E}_{CNN^{\Delta}}$ is very high for Orange, White, Blue, Magenta, Black and Red and is high for Yellow and Emerald Green. Instead, it is low for Purple. Color Deep has high values of F1-score for Orange, White, Blue, Red, Brown and Green while it has

Deep neural network	*Precision*	*Recall*	*F1-score*	*Accuracy*
$\mathcal{E}_{CNN^{\Delta}}$	**0.78**	**0.78**	**0.78**	**0.79**
CNN$^{\Delta}$	0.69	0.66	0.65	0.66
ResNet50v2	0.55	0.55	0.52	0.55
ResNet152v2	0.58	0.58	0.54	0.58
InceptionResNetV2	0.59	0.59	0.55	0.58
DenseNet169v2	0.58	0.55	0.53	0.55
DenseNet201v2	0.59	0.59	0.55	0.58
Color Deep (avg.)	0.50	0.62	0.60	0.63
Color Deep (max.)	0.65	0.66	0.62	0.66

Table 7: Results of $\mathcal{E}_{CNN^{\Delta}}$ and the other models chosen for comparison when the images of $\mathcal{D}^{-}$ are provided in input

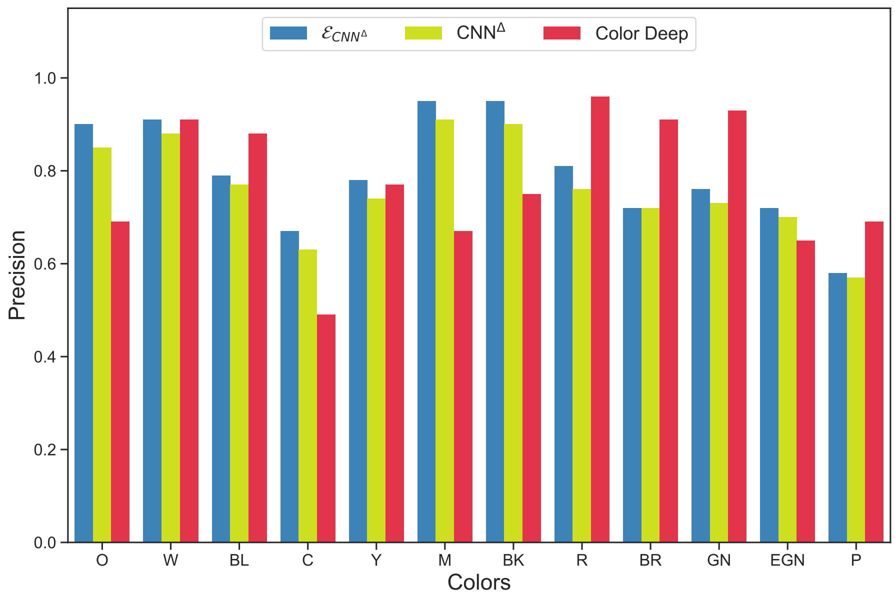

Figure 7: Values of Precision obtained by applying $\mathcal{E}_{CNN^{\Delta}}$, a single CNN$^{\Delta}$ and Color Deep

low values for Cyan. We also observe that Orange, White, Blue and Red show high values of F1-score for both approaches. Interestingly, there is no color that shows low values of F1-score for both $\mathcal{E}_{CNN^{\Delta}}$ and Color Deep. This result could be usefully exploited in future developments, as we will see below. Finally, the values of F1-score are always better for $\mathcal{E}_{CNN^{\Delta}}$ than for a single CNN$^{\Delta}$.

4.3 Discussion

As shown in Section 4.2, $\mathcal{E}_{CNN^{\Delta}}$ achieves very good results. These performances are reached thanks to two expedients, namely the difference method and ensemble learning. Working with fabrics is not easy because they often present textures, which can make color recognition very difficult. However, the two expedients mentioned above allow us to increase the amount of information provided in input to $\mathcal{E}_{CNN^{\Delta}}$, without the risk of overfitting it.

The important role played by the difference method and ensemble learning in achieving high performance is evident when comparing $\mathcal{E}_{CNN^{\Delta}}$ with other deep learning architectures. In fact, we

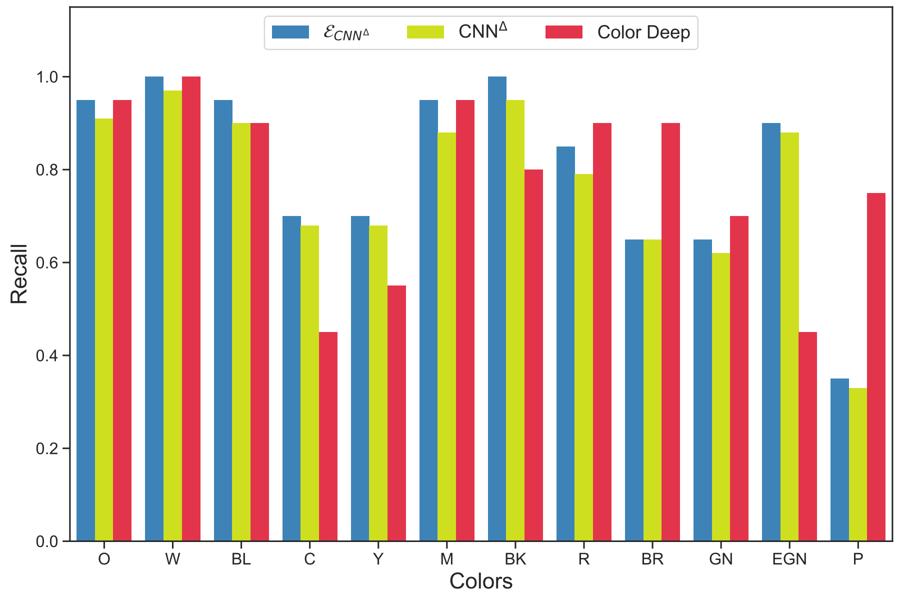

Figure 8: Values of Recall obtained by applying $\mathcal{E}_{CNN^{\Delta}}$, a single CNN$^{\Delta}$ and Color Deep

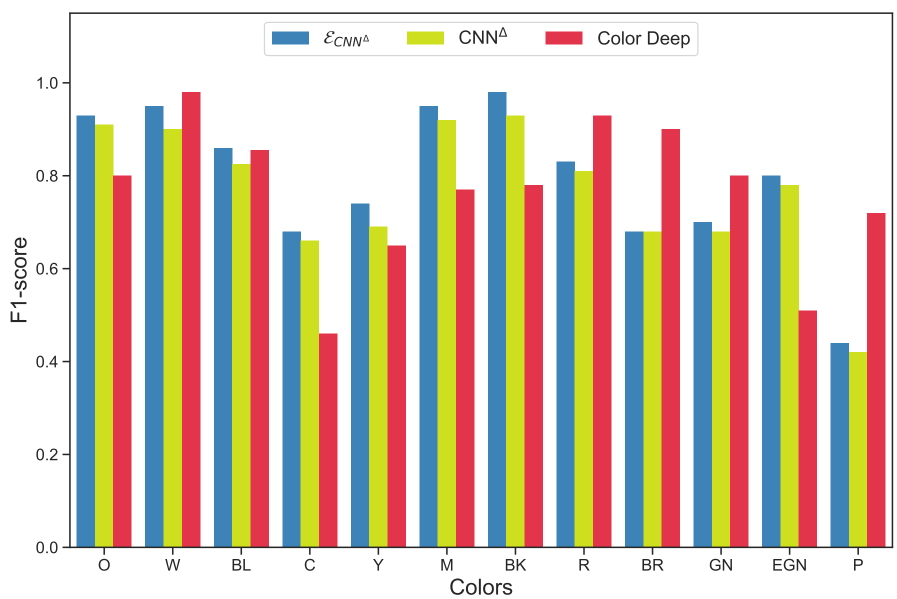

Figure 9: Values of F1-score obtained by applying $\mathcal{E}_{CNN^{\Delta}}$, a single CNN$^{\Delta}$ and Color Deep

have seen that the difference method allows a single CNN$^\Delta$ and $\mathcal{E}_{CNN^\Delta}$ to perform better than the other models. The usage of ensemble learning in $\mathcal{E}_{CNN^\Delta}$ allows it to perform better than a single CNN$^\Delta$. This shows that both the expedients used in the definition of $\mathcal{E}_{CNN^\Delta}$ help to achieve the goals we had set for ourselves.

Our approach has a high potential in several scenarios. For example, it could be integrated into a recommender system that matches and suggests clothes to create a pleasing outfit. In turn, this system could be integrated into an app that could suggest what to wear based on the clothes in the closet. In addition, it could be used to help color blind people to recognize the correct colors of clothes.

However, besides these merits, our approach has also some limitations, which represent the subject of future investigations. First, in this paper, we considered a small number of colors (i.e., 12) in our experiments. These are the most frequent ones present in fabric images, but they are a fraction of the colors that can be found in this kind of image and that we could use in future experiments and applications. It is clear that increasing the number of reference colors, and therefore the number of classes, makes the classification task much more complex. Already a classification with 12 possible classes is not very common in the machine learning context. A further increase in the number of classes is possible, but it requires much more effort and is subject to many more errors. It must be taken into account that, continuing in this direction, the efforts needed could increase very much and the performance might decrease. So, at some point, a cost-benefit analysis must be done and the suitable trade-off must be found.

Going into more detail in the examination of results, we have seen that $\mathcal{E}_{CNN^\Delta}$ has unsatisfactory performances on some colors (e.g., Purple and Cyan). This contributes to lower its overall performance but, more importantly, makes it not fully adequate in classifying fabrics whose color could fall into one of those classes in which it showed low performances.

5 Conclusion

In this paper, we have proposed a new approach for identifying fabric colors. We have seen that such a task is not trivial, because fabric images contain textural patterns that can make color classification hard. To address this issue, we have introduced a new color space, called difference space. Then, we have seen that this way of proceeding allows us to overcome the limitations of the previous representations and makes color recognition much more effective. Indeed, such an expedient allows us to provide much more information as input to the deep learning system.

Based on these results, we have designed a Convolutional Neural Network model, called CNN$^\Delta$, to operate in this new color space. To further improve the performance, we have used ensemble learning and we have combined three versions of CNN$^\Delta$ models to build an ensemble model called $\mathcal{E}_{CNN^\Delta}$. Then, we have carried out some experiments on the Fabrics Dataset [13] that showed how the adoption of the two expedients (difference method and ensemble strategy) allows $\mathcal{E}_{CNN^\Delta}$ to obtain very satisfactory results, better than those reached by other learning models already proposed in the literature.

This paper should not be considered an ending point but, rather, a starting point for further research efforts. First, as we said above, we plan to increase the number of reference colors used in the classification task. Then, we think to extend our model in such a way that it can assign more colors to a single fabric image. In fact, there are fabrics with several colors and, therefore, it would be important to have an approach capable of handling this situation. These two factors could make the problem of color identification in fabric images much more complex and could provide many interesting insights for researchers who want to address this issue in the future. Also, as we saw in Figure 9, there is some complementarity between CNN$^{\Delta}$ and Color Deep models. In fact, for the colors in which one of them reaches low values of F1-score, the other one obtains high values of the same parameter. This suggests us to verify the possibility to build an ensemble that includes CNN$^{\Delta}$ and Color Deep in order to exploit the strengths of both of them and reduce their weaknesses. Last but not least, if we had access to a much larger database than the Fabrics Dataset, it would be interesting to evaluate the application of a Vision Transformers based methodology [14, 37] for color classification of fabric images. In fact, we consider this architecture very promising, but it needs thousands of images for its application. It is our intention to try it in the future.

References

[1] B.S. Anami and M.C. Elemmi. A rule based approach for classification of shades of basic colors of fabric images. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(2):389–400, 2015.

[2] D.H. Apriyanti, L.J. Spreeuwers, P.J. Lucas, and R.N. Veldhuis. Automated color detection in orchids using color labels and deep learning. *PLoS ONE*, 16(10):e0259036, oct 2021.

[3] S. Basar, A. Adnan, N.H. Khan, and S. Haider. Color image segmentation using k-means classification on rgb histogram. *Recent Advances In Telecommunications, Informatics and Educational Technologies*, pages 257–262, 2014.

[4] D.S. Bloomberg. Color quantization using modified median cut. In *Leptonica*, pages 1–10. 2008.

[5] D. Borza, T. Ileni, and A. Darabant. A Deep Learning Approach to Hair Segmentation and Color Extraction from Facial Images. In *Proc. of the International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS'18)*, pages 438–449, Auckland, New Zealand, 2018. Springer.

[6] P. Chen, X. Bai, and W. Liu. Vehicle color recognition on urban road by feature context. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2340–2346, 2014. IEEE.

[7] Z. Di, Z. Luoqing, and S. Jun. A novel feedback error-correcting algorithm for automatic recognition of the color and weave pattern of yarn-dyed fabrics. *Textile Research Journal*, 83:1673 – 1689, 2013.

[8] A.K. Gupta and D.J. Bora. A novel color image segmentation approach based on k-means clustering with proper determination of the number of clusters and suitable distance metric. *International Journal of Computer Science & Engineering Technology*, 7(09):395–409, 2016.

[9] M. Hajiarbabi and A. Agah. Human skin color detection using neural networks. *Journal of Intelligent Systems*, 24(4):425–436, 2015. De Gruyter.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR'16)*, pages 770–778, Las Vegas, NV, USA, 2016.

[11] C. Hu, X. Bai, L. Qi, P. Chen, G. Xue, and L. Mei. Vehicle color recognition with spatial pyramid deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2925–2934, 2015. IEEE.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K.Q. Weinberger. Densely Connected Convolutional Networks. In *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pages 2261–2269, Honolulu, HI, USA, 2017. IEEE Computer Society.

[13] C. Kampouris, S. Zafeiriou, A. Ghosh, and S. Malassiotis. Fine-Grained Material Classification Using Micro-geometry and Reflectance. In *Proc. of the European Conference on Computer Vision (ECCV'16)*, pages 778–792, Amsterdam, The Netherlands, 2016. Springer.

[14] S. Khan, M. Naseer, M. Hayat, S.W. Zamir, F.S. Khan, and M. Shah. Transformers in vision: A survey. *ACM Computer Surveys*, 2021. ACM.

[15] D.K. Kim. Color Detection Using Gaussian Mixture Model. *Journal of Theoretical and Applied Information Technology*, 95(17):4313–4320, 2017.

[16] H.K. Kim, J.H. Park, and H.Y. Jung. An Efficient Color Space for Deep-Learning Based Traffic Light Recognition. *Journal of Advanced Transportation*, page 2365414, 2018.

[17] D.P. Kingma and J.L. Ba. Adam. A Method for Stochastic Optimization. In *Proc. of the International Conference on Learning Representations (ICLR'15)*, San Diego, CA, USA, 2015.

[18] C.F.J. Kou, C.Y. Shih, C.Y. Kao, and J.Y. Lee. Color and Pattern Analysis of Printed Fabric by an Unsupervised Clustering Method. *Textile Research Journal*, 75(1):9–12, 2005.

[19] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *Proc. of the International Conference on Computer Vision Workshops (ICCV'13)*, pages 554–561, Sydney, Australia, 2013.

[20] Z. Li, G. Golwala, S. Sundaram, and J. Allebach. Use of Color Information in the Analysis of Fashion Photographs. *Electronic Imaging*, 2018(10):446–1, 2018. Society for Imaging Science and Technology.

[21] B. Liu, J. Gan, B. Wen, Y. LiuFu, and W. Gao. An automatic coloring method for ethnic costume sketches based on generative adversarial networks. *Applied Soft Computing*, 98:106786, 2021. Elsevier.

[22] G.H. Liu and J.Y. Yang. Content-based image retrieval using color difference histogram. *Pattern Recognition*, 46(1):188–198, 2013. Elsevier.

[23] A.G. Oskouei, M. Hashemzadeh, B. Asheghi, and M.A. Balafar. CGFFCM: CLuster-weight and group-local feature-weight learning in fuzzy C-means clustering algorithm for color image segmentation. *Applied Soft Computing*, 113:108005, 2021. Elsevier.

[24] R.F. Rachmadi and I. Purnama. Vehicle Color Recognition using Convolutional Neural Network. *arXiv preprint*, abs/1510.07391, 2015.

[25] A. Rame, A. Douillard, and C. Ollion. CORE: Color Regression for Multiple Colors Fashion Garments. *ArXiv preprint*, abs/2010.02849, 2020.

[26] T. Sandhan and J. Y. Choi. Anti-Glare: Tightly Constrained Optimization for Eyeglass Reflection Removal. In *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pages 1675–1684, Honolulu, HI, USA, 2017.

[27] A. Shams-Nateri and E. Hasanlou. Computer vision techniques for measuring and demonstrating color of textile. In W.K. Wong, editor, *Applications of Computer Vision in Fashion and Textiles*, The Textile Institute Book Series, pages 189–220. Woodhead Publishing, 2018.

[28] J. Shen, H. Ma, W. Chen, and X. Zhou. A novel analysis of color component for top dyed melange yarn with support vector machine. *Color Research & Application*, 41(6):636–641, 2016.

[29] S.M. Singh and K. Hemachandran. Content-based image retrieval using color moment and Gabor texture feature. *International Journal of Computer Science Issues*, 9(5):299, 2012.

[30] B. Su, J. Shao, J. Zhou, X. Zhang, and L. Mei. Vehicle color recognition in the surveillance with deep convolutional neural networks. In *Proc. of the Joint International Mechanical, Electronic and Information Technology Conference (JIMET 2015)*, Chongqing, China, 2015.

[31] W. Sun, J.K. Hao, Y. Zang, and X. Lai. A solution-driven multilevel approach for graph coloring. *Applied Soft Computing*, 104:107174, 2021. Elsevier.

[32] C. Szegedy, L. Wei, J. Yangqing, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. of the International Conference on Computer Vision and Pattern Recognition (CVPR'15)*, pages 1–9, Boston, MA, USA, 2015.

[33] T. Vafeiadis, N. Kolokas, N. Dimitriou, A. Zacharaki, M. Yildirim, H. G. Selvi, D. Ioannidis, and D. Tzovaras. A comparison of 2DCNN network architectures and boosting techniques for regression-based textile whiteness estimation. *Simulation Modelling Practice and Theory*, 114:102400, 2022.

[34] M. Veluchamy and B. Subramani. Fuzzy dissimilarity color histogram equalization for contrast enhancement and color correction. *Applied Soft Computing*, 89:106077, 2020. Elsevier.

[35] P.S. Vitthal, S. Balasubramanian, and P.S. Mane. Color Analysis and Classification Based on Machine Learning Technique Using RGB Camera Industrial Practice and Experience Paper. In *Proc. of the International Conference on Computational Intelligence and Computing Research (ICCIC'17)*, pages 1–5, Coimbatore, India, 2017.

[36] L. Xie, Y. Shen, and X. Chen. Study on Automatic Recognition of Fabric Color and Matching to Standard Color Chip by Computer Vision and Image Analysis Technology. *Journal of Fiber Bioengineering and Informatics*, 9(1):29–37, 2016.

[37] Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, and C. Xu. Transformers in computational visual media: A survey. *Computational Visual Media*, 8(1):33–62, 2022. Springer.

[38] Z. Yang, L. Leng, and B.G. Kim. StoolNet for Color Classification of Stool Medical Images. *Electronics*, 8(12):1464, 2019.

[39] H. Ye, L. Zheng, and P. Liu. Color detection and segmentation of the scene based on Gaussian mixture model clustering. In *Proc. of the International Conference on Electronics Information and Emergency Communication (ICEIEC'17)*, pages 503–506, Shenzhen, China, 2017.

[40] S. Ye, J.L. Yin, B. Chen, D. Chen, and Y. Wu. Single Image Glare Removal Using Deep Convolutional Networks. In *Proc. of the International Conference on Image Processing (ICIP'20)*, pages 201–205, United Arab Emirates, 2020.

[41] Z. Zehra and M.N. Bashir. Color Fastness Grading System for Textile Industry Using CIEL*a*b Color Space. In *Proc. of the International Conference on Big Data and Smart City (ICBDSC'19)*, pages 1–5, Yokohama, Japan, 2019.

[42] J. Zhang, R. Pan, E. Gao, and D. Zhu. Automatic detection of layout of color yarns of yarn-dyed fabric. Part 1: Single-system-mélange color fabrics. *Color Research & Application*, 40(6):626–636, 2015.

[43] J. Zhang, R. Pan, W. Gao, and D. Zhu. Automatic recognition of the color effect of yarn-dyed fabric by the smallest repeat unit recognition algorithm. *Textile Research Journal*, 85:432 – 446, 2015.

[44] L. Zhang, C. Long, X. Zhang, and C. Xiao. RIS-GAN: Explore Residual and Illumination with Generative Adversarial Networks for Shadow Removal. In *Proc. of the International Conference on Artificial Intelligence (AAAI'20)*, volume 34, pages 12829–12836, New York, NY, USA, 2020.

[45] M. Zhang and Y. Jia. Color Recognition Model Based on Multiple SVMs for Bobbin Sorting Machine. In *Proc. of the International Symposium on Computational Intelligence and Design (ISCID'12)*, volume 2, pages 218–221, Washington DC, USA, 2012.

[46] M. Zhang, P. Wang, and X. Zhang. Vehicle Color Recognition Using Deep Convolutional Neural Networks. In *Proc. of the International Conference on Artificial Intelligence and Computer Science (AICS'19)*, pages 236—238, Wuhan, China, 2019. Association for Computing Machinery.

[47] Q. Zhang, L. Zhuo, J. Li, J. Zhang, H. Zhang, and X. Li. Vehicle color recognition using Multiple-Layer Feature Representations of lightweight convolutional neural network. *Signal Processing*, 147:146–153, 2018.

[48] Z. Zhuang, Y. Liu, F. Ding, and Z. Wang. Online Color Classification System of Solid Wood Flooring Based on Characteristic Features. *Sensors*, 21(2):336, 2021.