Bag of Color Features for Color Constancy

Firas Laakom, Nikolaos Passalis, Jenni Raitoharju, *Member, IEEE,* Jarno Nikkanen, Anastasios Tefas, *Member, IEEE*, Alexandros Iosifidis, *Senior Member, IEEE*, and Moncef Gabbouj, *Fellow, IEEE*

***Abstract*— In this paper, we propose a novel color constancy approach, called Bag of Color Features (BoCF), building upon Bag-of-Features pooling. The proposed method substantially reduces the number of parameters needed for illumination estimation. At the same time, the proposed method is consistent with the color constancy assumption stating that global spatial information is not relevant for illumination estimation and local information (edges, etc.) is sufficient. Furthermore, BoCF is consistent with color constancy statistical approaches and can be interpreted as a learning-based extension of many statistical approaches. To further improve the illumination estimation accuracy, we propose a novel attention mechanism for the BoCF model with two variants based on self-attention. BoCF approach and its variants achieve competitive, compared to the state of the art, results while requiring much fewer parameters on three benchmark datasets: ColorChecker RECommended, INTEL-TUT version 2, and NUS8.**

***Index Terms*— Color constancy, illumination estimation, bag of features, attention mechanism.**

I. Introduction

COLOR constancy in general is the ability of an imaging system to discount the effects of illumination on the observed colors in a scene [1], [2]. When a person stands in a room lit by a colorful light, the Human Visual System (HVS) unconsciously removes the lightening effects and the colors are perceived as if they were illuminated by a neutral, white light. While this ability is very natural for the HVS, mimicking the same ability in a computer vision system is a challenging and under-constrained problem. Given a green pixel, one can not assert if it is a green pixel under a white illumination or a white pixel lit with a greenish illumination. Nonetheless, illumination estimation is considered an important component of many higher level computer vision tasks such as object recognition and tracking. Thus, it has been extensively studied in order to develop reliable color constancy systems which can achieve illumination invariance to some extent [1], [3].

The RGB image value $\rho(x, y)$ in the position (x, y) of an image can be expressed as a function depending on three key factors [3]: the illuminant distribution $I(x, y, \lambda)$, the surface reflectance $R(x, y, \lambda)$ and the camera sensitivity $S(\lambda)$, where λ is the wave length. This dependency is expressed as

$$\rho(x, y) = \int_{\lambda} I(x, y, \lambda) R(x, y, \lambda) S(\lambda) d\lambda. \tag{1}$$

Color constancy methods [3], [4] aim to estimate a uniform projection of $I(x, y, \lambda)$ on the sensor spectral sensitivities $S(\lambda)$, i.e.,

$$I = I(x, y) = \int_{\lambda} I(x, y, \lambda) S(\lambda) d\lambda, \tag{2}$$

where I is the global illumination, i.e., it is assumed constant over the scene.

Recently, deep learning approaches and Convolutional Neural Networks (CNNs) in particular have become dominant in almost all computer vision tasks, including color constancy [5]–[8], due to their ability to take raw images directly as input and incorporate feature extraction in the learning process [9]. Despite their accuracy in estimating illumination across multiple datasets [6], [10], [11], deploying CNN-based approaches on low computational power devices, e.g., mobile devices, is still limited, since most of the high-accuracy deep models are computationally expensive [6]–[8], which make them inefficient in terms of time and energy consumption. Additionally, most of the available datasets for illumination estimation are rather small-scale [10], [12], [13] and hence not suitable for training large models. For this purpose, many state of the art approaches [5], [6] rely on pre-trained networks to overcome this limitation. On the other hand, these pre-trained networks [9], [14] are originally trained for a classification task. Thus, they are usually agnostic to the illumination color. This makes their usage in color constancy counter-intuitive as the illumination information is distorted in the early pre-trained layers. An alternative approach is of course to reduce the number of model parameters in order to use existing datasets, as shallower models, in general, need less examples to learn.

Manuscript received June 15, 2019; revised December 11, 2019, February 29, 2020, and April 14, 2020; accepted June 12, 2020. Date of publication July 1, 2020; date of current version July 14, 2020. This work was supported in part by the NSF-Business Finland Center for Visual and Decision Informatics Project (CVDI) and in part by the Project AMALIA, through the Intel Finland under Grant 3333/31/2018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yun Fu. *(Corresponding author: Firas Laakom.)*

Firas Laakom, Nikolaos Passalis, Jenni Raitoharju, and Moncef Gabbouj are with the Faculty of Information Technology and Communication Sciences, Tampere University, FI-33014 Tampere, Finland (e-mail: firas.laakom@tuni.fi; nikolaos.passalis@tuni.fi; jenni.raitoharju@tuni.fi; moncef.gabbouj@tuni.fi).

Jarno Nikkanen is with INTEL, FI-33720 Tampere, Finland (e-mail: jarno.nikkanen@intel.com).

Anastasios Tefas is with the Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece (e-mail: tefas@aiia.csd.auth.gr).

Alexandros Iosifidis is with the Department of Engineering, Aarhus University, 8200 Aarhus, Denmark (e-mail: alexandros.iosifidis@eng.au.dk).

Digital Object Identifier 10.1109/TIP.2020.3004921

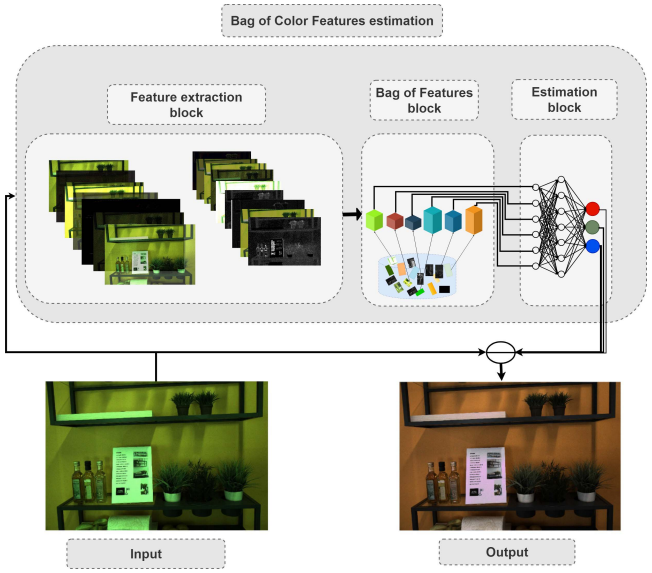

Fig. 1. Building blocks of Bag of Color Features (BoCF) approach for illumination estimation.

Furthermore, in [13], [15] it is argued that global spatial information is not an important feature in color constancy. The local information, i.e., the color distribution and the color gradient distribution (i.e. edges) can be sufficient to extract the illumination information [13]. Thus, using regular neural networks configurations to extract deep features is counter-intuitive in this particular problem. To address these drawbacks and challenges, we propose in this paper a novel color constancy deep learning approach called Bag of Color Features (BoCF). BoCF uses Bag-of-Features Pooling [16], which takes advantage of the ability of CNNs to learn relevant shallow features while keeping the model suitable for low-power hardware. Furthermore, the proposed approach is consistent with the assumption that global spatial information is not relevant [13], [15] for color illumination estimation.

Bag-of-Features Pooling is a neural extension [16], [17] of the famous Bag-of-Features model (BoF), also known as Bag-of-Visual Words (BoVW) [18], [19]. BoFs are widely used in computer vision tasks, such as action recognition [20], object detection/recognition, sequence classification [21], and information retrieval [22]. A BoF layer can be combined with convolutional layers to form a powerful convolutional architecture that is end-to-end trainable using the regular back-propagation algorithm [17].

The block diagram of the proposed BoCF model is illustrated in Figure 1. It consists of three main blocks: feature extraction block, Bag of Features block, and an estimation block. In the first block, regular convolutional layers are used to extract relevant features. Inspired by the assumption that second order gradient information is sufficient to extract the illumination information [13], we use only two convolutional layers to extract the features. In our experiments, we also study and validate this hypothesis empirically. In the second block, i.e., the Bag of Features block, the network learns the dictionary using back-propagation [17] over the non-linear transformation provided by the first block. This block outputs a histogram representation, which is fed to the last component, i.e., the estimation block, to regress to the scene illumination.

In most CNN-based approaches used to solve the color constancy problem [5]–[8], fully connected layers are connected directly to a flattened version of the last convolutional layer output. This increases the numbers of parameters dramatically, as convolutional layer outputs usually have a high dimensionality. In the proposed method, we address this problem by introducing an intermediate pooling block, i.e., the Bag of Features block, between the last convolutional layer and the fully connected layers. The proposed model achieves comparable results to previous state of the art illumination estimation methods while substantially reducing the number of the needed parameters, by up to 95%. Additionally, the pooling process natively discards all global spatial information, which is, as discussed earlier, irrelevant for color constancy. Using only two convolutional layers in the first block limits the model to only shallow features. These two advantages make the proposed approach both consistent and in full corroboration with statistical approaches [13].

To further improve the performance of the proposed model, we also propose two variants of a self-attention mechanism for the BoCF model. In the first variant, we add an attention mechanism between the feature extraction block and the Bag of Features block. This mechanism allows the network to dynamically select parts of the image to use for estimating the illumination, while discarding the remaining parts. Thus, the network becomes robust to noise and irrelevant features. In the second variant, we add an attention mechanism on top of the histogram representation, i.e., between the Bag of Features block and the estimation block. In this way, we allow the network to learn to adaptively select the elements of the histogram which best encode the illuminant information. The model looks over the whole histogram after the spatial information has been discarded and generates a proper representation according the current context (histogram). The introduced dynamics will be shown in the experiments to enhance the model performance with respect to all evaluation metrics and across all the datasets.

The main contributions of the paper are as follows:

- From the application side, we propose a novel CNN-based color constancy method, called Bag of Color Features (BoCF), which is light weight and able to achieve comparable results across multiple datasets compared to the state of the art.
- From the interpretability side, we establish explicit links between BoCF and prior statistical methods for illumination estimation and show that the proposed method can be framed as a learning-based extension of many statistical approaches. Thus, this powerful approach fills the gap and provides the missing links between CNN-based approaches and static approaches.
- From the methodology side, we propose two novel attention mechanisms for BoCF that can further boost the performance of neural BoF compared to the standard BoF in the color constancy problem. To the best of our

knowledge, this is the first work which combines an attention mechanism with Bag-of-Features Pooling.
- The proposed method is extensively evaluated over three datasets showing a competitive performance with respect to the existing state of the art, while substantially reducing the number of parameters.

The rest of this paper is organized as follows. Section II provides the background of color constancy approaches and a brief review of the Bag-of-Features Pooling technique and the attention mechanism used in this work. Section III details the proposed approach along with the two attention mechanisms based variants. Section IV introduces the datasets and the evaluation metrics used in this work along with the evaluation procedure. Section V presents the experimental results on three datasets: ColorChecker RECommended [12], NUS8-Dataset [13], and INTEL-TUT version2 [10]. In Section VI, we highlight the links between our approach and many existing methods and we show how our approach can be considered as a generic framework for expressing existing approaches. Section VII concludes the paper.

II. Related Work

A. Color Constancy

Typically, two types of color constancy approaches are distinguished, namely static methods and supervised methods. The former involves methods with static parameters settings that do not need any labeled image data for learning the model, while the latter are data-driven approaches that learn to estimate the illuminant in a supervised manner using labeled data.

1) Static Methods: Static methods exploit the statistical or physical properties of a scene by making assumptions about the nature of colors. They can be classified into two categories: methods based on low-level statistics [23]–[26] and methods based on the physics-based dichromatic reflection model [4], [15], [27], [28]. A number of approaches belonging to the first category were unified by Van de Weijer *et al.* [25] into a single framework, where the illumination I^{est} is estimated as follows:

$$I^{est}(n, p, \sigma) = \frac{1}{k}\left(\int_x \int_y |\bigtriangledown^n \rho_\sigma(x, y)|^p dx dy\right)^{\frac{1}{p}}, \quad (3)$$

where n denotes the derivative order, p the Minkowski norm and k the normalization constant for I^{est}. Also, $\rho_\sigma(x, y) = \rho(x, y) * g_\sigma(x, y)$ denotes the image convolution with a Gaussian filter with a scale parameter σ. This framework allows for deriving different algorithms simply by setting the appropriate values for n, p and σ. The well-known Gray-World method [24], corresponding to ($n = 0, p = 1$, $\sigma = 0$), assumes that under a neutral illumination the average reflectance in a scene is achromatic and the illumination is estimated as the shift of the image average color from gray. White-Patch [23] ($n = 0, p = \infty, \sigma = 0$), assumes that the maximum values of RGB color channels are caused by a perfectly reflecting surface in the scene. Therefore, the illumination components correspond to these maximum values. Besides Gray-World and White-Patch methods, which make use of the color distribution in the scene to build their estimations, Gray-Edge method [25] utilizes image derivatives. Instead of the global average color, Gray-Edge methods ($n = 1$, $p = p, \sigma = \sigma$) assume that the average color of edges or the gradient of edges is gray. The illuminant's color is then estimated as the shift of the average edge color from gray.

Physics-based dichromatic reflection models estimate the illumination by analyzing the scene and exploiting the physical interactions between the objects and the illumination. The main assumption of most methods in this category is that all pixels of a surface form a plane in RGB color space. As a scene contains multiple surfaces, this results in multiple planes. The intersection between these planes is used to compute the color of the light source [27]. Lee *et al.* [15] exploited the bright areas in the captured scene to obtain an estimate of the illuminant color. In this work, we establish links between our proposed approach, BoCF, and several static methods. We show that BoCF can be interpreted as a learning-based extension of several of these approaches.

2) Supervised Methods: Supervised methods can be further divided into two main categories: characterization-based methods [29], [30] and training-based methods [5], [6], [31], [32]. The former involves 'light' training processes in order to learn the characterization of the camera response in some way, while the latter involves methods that try to learn the illumination directly from the scene.

Gamut Mapping [29], [30] is one of the most famous characterization-based approaches. It assumes that, for a given illumination condition, only a limited number of colors can be observed. Thus any unexpected variation in the observed colors is caused by the light source illuminant. The set of colors that can occur under a given illumination, called canonical gamut, is first learned in a supervised manner. In the evaluation, an input gamut which represents the set of colors used to acquire the scene is constructed. The illumination is then estimated by mapping this input gamut to the canonical gamut. Fast Fourier Color Constancy (FFCC) method [33] reformulated the color constancy problem as a spatial localization task in the Fourier domain.

Another group of training-based methods combines different illumination estimation approaches and learns a model that uses the best performing method or a combination of methods to estimate the illuminant of each input based on the scene characteristics [31]. Bianco *et al.* used indoor/outdoor classification to select the optimal color constancy algorithm given an input image [32]. Lu *et al.* proposed an approach which exploits 3D scene information for estimating the color of a light source [34]. However, these methods tend to overfit and fail to generalize to all scene types.

The first attempt to use Convolutional Neural Networks (CNNs) for solving the illuminant estimation problem was established by Bianco *et al.* in [5], where they adopted a CNN architecture operating on small local patches to overcome the data shortage. In the testing phase, a map of local estimates is pooled to obtain one global illuminant estimate using median or mean pooling. Hu *et al.* [6] introduced a pooling layer, namely confidence-weighted pooling. In their fully convolutional network, they incorporate learning the

confidence of each patch of the image in an end-to-end learning process. Patches in an image can carry different confidence weights according to their estimated accuracy in predicting the illumination. Shi *et al.* [7] proposed a network with two interacting sub-networks to estimate the illumination. One sub-network, called the hypothesis network, is used to generate multiple plausible illuminant estimations depending on the patches in the scene. The second sub-network, called the selection network, is trained to select the best estimate generated by the first sub-network. Inspired by the success of Generative Adversarial Networks (GANs) in image to image translation [35], Das *et al.* formulated the illumination estimation task as an image-to-image translation task [36] and used a GAN to solve it. However, these CNN-based methods suffer from certain weaknesses: computational complexity and disconnection with both the illumination assumption [13] and the prior static methods, e.g., Grey-World [24] and White-Patch [23]. This paper attempts to cure these drawbacks by proposing a novel CNN approach, BoCF, which discards the global spatial information in agreement with [13] and [25], and is competitive with the training-based methods while using only 5% of the parameters.

B. Bag-of-Features Pooling

Passalis and Tefas proposed a Bag-of-Features Pooling (BoFP) layer [16], [17], which is a neural extension of the Bag-of-Features model (BoF). BoFPL can be combined with convolutional layers to form a powerful architecture which can be trained end-to-end using the regular back-propagation algorithm [17], [37]. In this work, we use this pooling technique to learn the codebook of color features. Thus, the naming Bag of Color Features (BoCF). This pooling discards all the global spatial information and outputs a fixed length histogram representation. This allows us to reduce the large number of parameters usually needed when linking convolutional layers to fully connected layers. Furthermore, discarding global spatial information forces the network to learn to extract the illumination without global spatial inference, thus improving model robustness and adhering to the illumination assumption [13]. As an additional novel feature to the prior works using Bag-of- Features Pooling [17], [37], we propose an attention mechanism to enable the model to discard noise and focus only on relevant parts of the input presentation. To the best of our knowledge, this is the first work which combines attention mechanisms with Bag-of-Features Pooling.

C. Attention Mechanisms

Attention mechanisms were introduced in Natural Language Processing (NLP) [38] for sequence-to-sequence (seq2seq) models in order to tackle the problem of short-term memory faced in machine translators. They allow a machine translator to see the full information contained in the original input and then generate the proper translation for the current word. More specifically, they allow the model to focus on local or global features, as needed. Self-attention [39], also known as intra-attention, is an attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence. In other words, the attention mask is computed directly from the original sequence. This idea has been exported to many other problems in NLP and computer vision such as machine reading [40], text summarization [41], [42], and image description generation [43]. In [43], self-attention is applied to an image to enable the network to generate an attention mask and focus on the region of interest in the original image.

Attention in deep learning can be broadly interpreted as a mask of importance weights. In order to evaluate the importance of a single element, such as a pixel or a feature in general, for the final inference, one can form an attention vector by estimating how strongly the element is correlated with the other elements and use this attention vector as a mask when evaluating the final output [43]. Let $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbb{R}^n$ be a vector. The goal of a self-attention mechanism is to learn to generate a mask vector $\mathbf{v} \in \mathbb{R}^n$ depending only on $\mathbf{x}$, which encodes the importance weights of the elements of $\mathbf{x}$. Let f be a mapping function between $\mathbf{x}$ and $\mathbf{v}$. The dependency can be expressed as follows:

$$\mathbf{v} = f(\mathbf{x}) = [v_1, \ldots, v_n], \tag{4}$$

under the constraint:

$$\sum_{i=1}^{n} v_i = 1. \tag{5}$$

After computing the mask vector $\mathbf{v}$, the final output of the attention layer $\mathbf{y}$ is computed as follows:

$$\mathbf{y} = [y_1, \ldots, y_n] = [x_1 v_1, \ldots, x_n v_n]. \tag{6}$$

The concept of attention, i.e., focusing on particular regions to extract the illumination information in color constancy, can be rooted back to many statistical approaches. For example, White-Patch reduces this region to the pixel with the highest RGB values. Other methods, such as [15] focus on the bright areas in the captured scene, called specular highlights. Instead of making such a strong assumption on the relevant regions, in BoCF we allow the model to learn to extract these regions dynamically. In FC4 [6], the concept of confidence maps resembles attention, as the network learns to assign a confidence score to each patch. To the best of our knowledge, this is the first work, which directly uses attention mechanisms in the color constancy problem.

III. Proposed Approach

In order to reduce the number of parameters needed to learn the illumination [6], [7], we propose a novel color constancy approach based on the Bag-of-Features Pooling [17], called herein the BoCF approach. The proposed approach along with the novel attention variants is illustrated in Figure 2. The proposed model has three main blocks, namely the feature extraction, the Bag of Features, and the illumination estimation blocks. In the first block, a nonlinear transformation of a raw image is obtained. In the second block, a histogram representation of this transformation is compiled. This histogram is used in the third block to estimate the illumination. For the first variant of attention, as illustrated in Figure 2 (in red),

the attention is applied on the image representation before the BoF, whereas for the second variant (in green) it is applied on the histogram output of the BoF.

A. Feature Extraction

The feature extraction algorithm takes a raw image as an input and outputs a nonlinear transformation representing the image features. A CNN is used in this block. CNNs are known for their ability to extract relevant features directly from raw images. Technically, any CNN architecture can be used in this block. However, we observed in our experiments that only two convolutions followed by downsampling layers, e.g., max-pooling yields satisfactory results. This is in accordance with the assumption of statistical methods that the second order information is enough to estimate the illumination [13], [25].

After a raw image is fed to the feature extraction block, the output of the last convolutional layer is used to extract feature vectors that are subsequently fed to the next block. The number of extracted feature vectors depends on the size of the feature map and the used filter size as described in [17].

B. Bag-of-Features

The Bag-of-Features is essentially a codebook (dictionary) learning component. The output features of the previous block are pooled using the Bag-of-Features Pooling and mapped to a final histogram representation. During training, the network optimizes the codebook using the traditional back-propagation. The output of this block is a histogram of a fixed size, i.e., the size of the codebook, which is a hyper-parameter that needs to be tuned to avoid over-fitting. This approach discards all global spatial information. As described in [17], the Bag-of-Features Pooling is composed of two sub-layers: an RBF layer that measures the similarity of the input features to the RBF centers and an accumulation layer that builds the histogram of the quantized feature vectors. The normalized output of each RBF neuron can be expressed as

$$[\Phi(\mathbf{x})]_k = \frac{exp(-||\mathbf{x} - \mathbf{y}_k||/m_k)}{\sum_j exp(-||\mathbf{x} - \mathbf{y}_j||/m_j)}, \tag{7}$$

where $\mathbf{x}$ is a feature vector (output of the feature extraction block), $\mathbf{y}_k$ is the center of the k-th RBF neuron, *exp* is the exponential function, and m_k is a scaling factor. The output of the RBF neurons is accumulated in the next layer, compiling the final representation $\mathbf{s}$ of each image:

$$\mathbf{s} = \frac{1}{N} \sum_j \Phi(\mathbf{x}_j), \tag{8}$$

where N is the number of feature vectors extracted from the last convolutional layer.

C. Illumination Estimation

In the illumination estimation block, we use a fully connected network, which takes as input the histogram formed in the previous block and outputs the illumination, i.e, 3-element vector containing an RGB value. To this end, we use a multi-layer perceptron with only one hidden layer.

Let $\mathbf{s} \in \mathbb{R}^n$ be the histogram compiled by the second block, the BoF block, as defined in Eq. 8. The intermediate layer output $\mathbf{h} \in \mathbb{R}^m$ of the illumination estimation block can be computed as follows

$$\mathbf{h} = \varphi(\mathbf{W}_1\mathbf{s} + \mathbf{b}_1), \tag{9}$$

where $\mathbf{W}_1 \in \mathbb{R}^{n \times m}$ is the weight matrix, $\mathbf{b}_1 \in \mathbb{R}^m$ is the bias vector, and φ is the Rectified Linear Units (ReLU) activation function [44]. The final estimate $\mathbf{I} \in \mathbb{R}^3$ is computed as follows

$$\mathbf{I} = \phi(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2), \tag{10}$$

where $\mathbf{W}_2 \in \mathbb{R}^{m \times 3}$ is the weight matrix, $\mathbf{b}_2 \in \mathbb{R}^3$ is the bias vector, and ϕ is the *softmax* activation function defined by

$$\phi(a_i) = \frac{exp(a_i)}{\sum_j exp(a_j)}, \tag{11}$$

D. Attention Mechanism for BoCF

We introduce a novel attention mechanism in the BoCF model to enable the algorithm to dynamically learn to focus on a specific region of interest in order to yield a confident output. We combine self-attention, described in Section II-C, with the Bag-of-Features Pooling for the color constancy problem. We propose two variants of this mechanism which can be applied in our model.

In the first variant, we apply attention on the nonlinear transformation of the image after the feature extraction block. This enables the model to learn to 'attend' the region of the interest in the mapping and to reduce noise before pooling. By applying attention in this stage, the number of parameters will rise exponentially as we need as many parameters as features.

In the second variant, we apply the attention mechanism on the histogram representation of the BoCF, i.e., after the global spatial information has been discarded. This enables the model to dynamically learn to 'attend' to the relevant parts of the histogram which encode the illuminant information. In this variant, the attention mask size is equal to the size of the histogram. Thus, the number of additional parameters is relatively small.

Following the notations of Eq. 4 and Eq. 5, $\mathbf{x} \in \mathbb{R}^n$ is the feature vector extracted from the CNN for the first mechanism or the histogram output for the second mechanism. The attention mask $\mathbf{v} \in \mathbb{R}^n$ is obtained via the following transformation:

$$\mathbf{v} = \phi(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{12}$$

where $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a weight matrix, $\mathbf{b} \in \mathbb{R}^N$ is the bias.

Using *softmax* as ϕ ensures that the masking constraint defined in (Eq. 5) is not violated. Finally, $\mathbf{y}$, the final output of the attention mechanism, is computed using the following equation

$$\mathbf{y} = \lambda(\mathbf{v} \odot \mathbf{x}) + (1 - \lambda)\mathbf{x}, \tag{13}$$

where $\odot$ is the element wise product operator and $\lambda \in \mathbb{R}$ is a weighting parameter between the original $\mathbf{x}$ and the masked $\mathbf{x}$, i.e., $\mathbf{v} \odot \mathbf{x}$. λ is a learnable parameter in our model. Not using

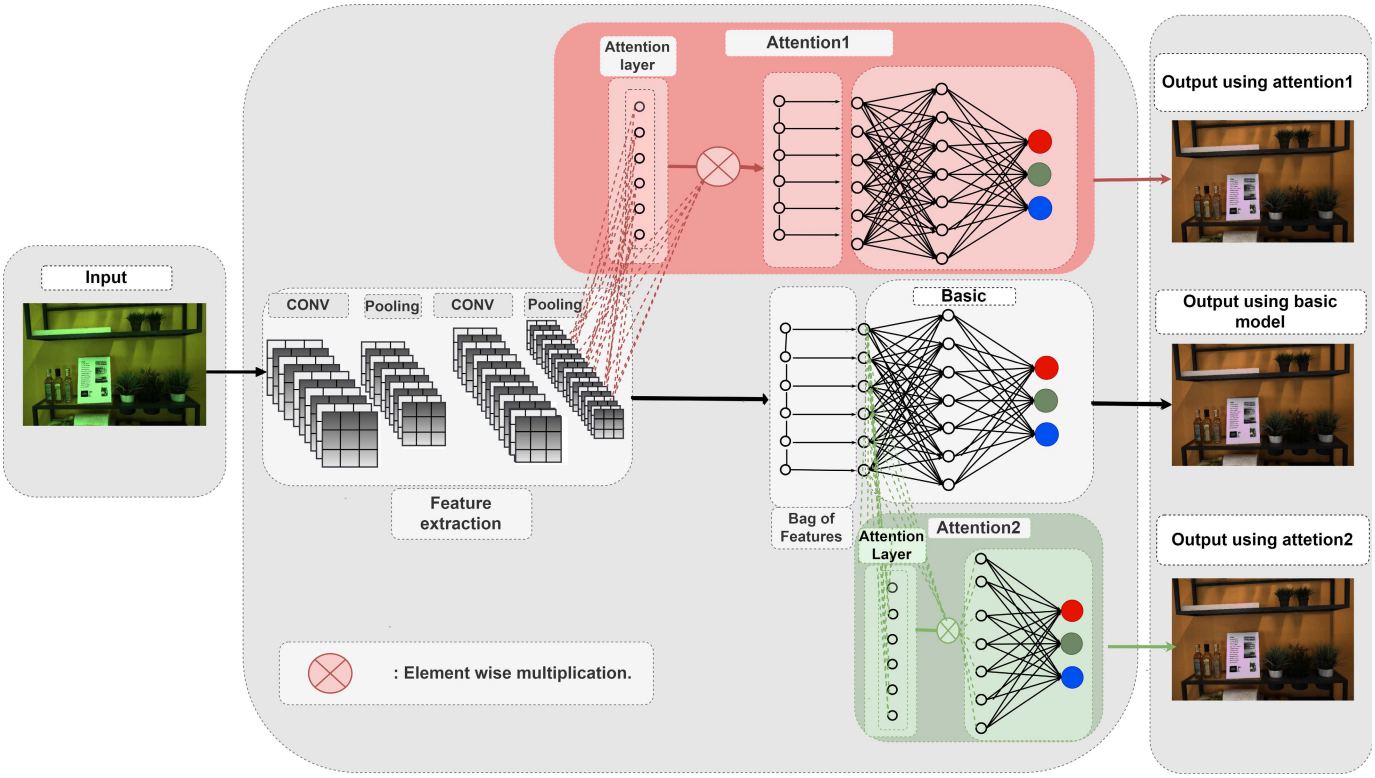

Fig. 2. Proposed approach (basic, no attention) along with attention variants (Attention1 in red and Attention2 in green). An example input image is fed into the three different models and the corresponding three different outputs for the estimated illuminant are illustrated in the Figure.

λ and outputting only the masked histogram is also an option. However, we determined experimentally that outputting the weighted sum of both the original and the masked version is more robust and stable for the gradient-based optimizers, since it is less susceptible to random initialization weights of the attention.

Parameter λ can be optimized using the gradient decent in the back-propagation process along with the rest of the parameters. Its gradient with respect to the output of the attention block can be obtained via the following equation

$$\frac{\partial \mathbf{y}}{\partial \lambda} = \mathbf{v} \odot \mathbf{x} - \mathbf{x}. \tag{14}$$

IV. Experimental Setup

In this section, we present the experimental setup used in this work. In Subsection IV-A, we introduce the datasets used to test our models. In Subsection IV-B, we report the network architectures of the three blocks used in BoCF. In Subsection IV-C, we detail the evaluation process followed in our experiments. Finally, the evaluation metrics used are briefly described in Subsection IV-D.

A. Image Datasets

1) ColorChecker RECommended Dataset: ColorChecker RECommended dataset [12] is a publicly available updated version of Gehler-Shi dataset [11][1] with a proposed (recommended) ground truth to use for evaluation. This dataset contains 568 high-quality indoor and outdoor images acquired by two cameras: Canon 1D and Canon 5D. Similar to the works in [5]–[8], for ColorChecker REComended dataset, we used three-fold cross validation to evaluate our algorithms.

2) NUS-8 Camera Dataset: NUS-8 is a publicly available dataset,[2] containing 1736 raw images from eight different camera models. Each camera has about 210 images. Following previous works [6], [13], we perform 3-fold cross validation experiments on each camera separately and report the mean of all the results for each evaluation metric. As a result, although the total number of images in NUS-8 dataset is large, each experiment involves using only 210 images for both training and testing.

3) INTEL-TUT2: INTEL-TUT2[3] is the second version of the publicly available INTEL-TUT dataset [10]. The main particularity of this dataset is that it contains a large number of images taken by several cameras from different scenes. We use this dataset with an extreme testing protocol, the third protocol described in [10]. The models are trained with images acquired by one camera and containing one type of scene and tested on the other cameras and the other scenes. This extreme test is useful to show the robustness of a given model and its ability to generalize across different cameras and scenes.

INTEL-TUT2 contains images acquired with three different cameras, namely Canon, Nikon, and, Mobile. For each camera, the images are divided into four sets: *field* (144 images per camera), *lab printouts* (300 images per camera), *lab real*

[1]http://www.cs.sfu.ca/ colour/data/shi_gehler/

[2]http://cvil.eecs.yorku.ca/projects/public_html/illuminant/illuminant.html

[3]http://urn.fi/urn:nbn:fi:csc-kata20170901151004490662

Fig. 3. Samples from INTEL-TUT2 dataset. The rows contain samples images taken by Canon, Nikon, and mobile cameras, respectively, while the columns contain images from *lab printouts*, *lab real* scenes, and *field*, respectively.

Fig. 4. Samples from *field2* set specific for Canon in INTEL-TUT2 dataset.

scenes (4 images per camera), and *field2*. The last set *field2* concerns only Canon and it has a total of 692 images. Figure 3 shows some samples from the *field*, *lab printouts*, and *lab real scenes* sets of the three cameras, while Figure 4 displays samples from *field2* related to Canon camera.

We used only Canon *field2* set for training and validation (80% for training and 20% for validation). We constructed two test sets. The first one, called *field* in this work, contains all the field images (field sets, i.e., non lab images) taken by the other camera models, i.e., Nikon and Mobile. The second set, called *non-field* in this work, contains the rest of the images, i.e., lab scenes, acquired by Nikon and Mobile (lab printouts and lab real scenes sets). Comparing the performance on these two sets allowed us to test both scene and camera invariance of the model. As we were using different camera models in same experiments, the variation of camera spectral sensitivity was discounted. For this purpose, we used Color Conversion Matrix (CCM) based preprocessing [45] to learn the 3×3 color conversion matrices (CCMs) for each camera pair.

B. Network Architectures

The BoCF network is composed of three blocks: the feature extraction, the Bag of Features block, and the illumination estimation blocks as described in Section III. The feature extraction block consists of convolution layers followed by max pooling operators. We experimented with different number of layers two and three. Thirty convolution filters of size 4×4 are used in each layer. Max-pooling with a window size 2 was applied in each layers. For the codebook size, i.e., number of RBF neurons in the Bag of Features block, we experimented with three different values 50, 150, and 200. The illumination estimation block consists of 2 fully connected layers. The first (hidden layer) has a size of 40 and it takes as an input the histogram representation and the second one (final output) has a size of 3 to output the illumination. The network is trained from scratch. The biases were initialized with zeroes while the convolution filters and the fully connected layers' weights with Xavier uniform. Figure 5 represents the architecture of the basic model used in our work with two convolution layers and a dictionary with 150 elements.

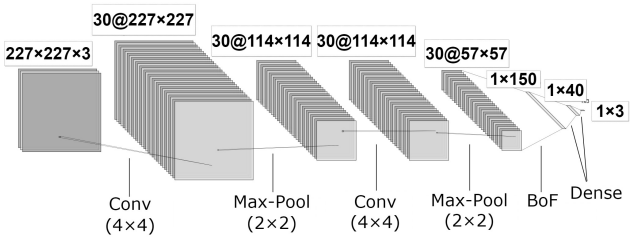

Fig. 5. Bag of Color Features (BoCF) network topology.

C. Evaluation Procedure

To evaluate the proposed approach, we used two sets of experiments. In the first set, we evaluated different variants of the model to study the effect of the hyper-parameters and validate the effectiveness of each component in our model by conducting ablation studies. For this purpose, we used ColorChecker RECommended dataset. In the second set of experiments, we compared our approach with current state-of-the-art approaches on the three datasets.

For all testing scenarios, we augmented the datasets using the following process: As the size of the original raw images is high, we first randomly cropped 512×512 patches of each image. This ensured getting meaningful patches. The crops were then rotated by a random angle between $-30°$ and $+30°$. Finally, we rescaled the RGB values of each patch and its corresponding ground truths by a random factor in the range of [0.8, 1.2]. Before feeding the sample to the network, we down-sampled it to 227×227. In testing, the images were resized to 227×227 to fit the network model.

Our network was implemented in Keras [46] with Tensorflow backend [47]. We trained our network end-to-end by back-propagation. The loss function used is the Recovery Angular Error defined in Sec IV-D. For optimization, Adam [48] was employed with a batch size of 15 and a learning rate of 3×10^{-4}. The model was trained on image patches of size 227×227 for 3000 epochs. The centers of the dictionary were initialized using the k-means algorithm as described in [17]. The parameter λ discussed in Section III-D was initialized as 0.5.

D. Evaluation Metrics

We report the mean of the top 25%, the mean, the median, Tukey's trimean, and the mean of the worst 25% of the recovery angular error (RAE) [49] between the ground truth

TABLE I
COMPARISON OF DIFFERENT VARIANTS OF THE PROPOSED BOCF APPROACH ON COLORCHECKER RECOMMENDED DATASET

Method	# par	Best 25%	Mean	Med.	Tri.	Worst 25%
BoCF (2conv+50 words + no attention)	13k	0.4	2.2	1.6	2.0	5.3
BoCF (2conv+150 words + no attention)	20k	0.3	2.1	1.5	1.6	5.1
BoCF (2conv+200 words + no attention)	23k	0.3	2.0	1.5	1.6	5.2
BoCF (3conv+150 words+ no attention)	37k	0.3	2.2	1.4	1.8	5.1
BoCF (4conv+150 words+ no attention)	52k	0.5	2.4	1.6	1.7	5.2
BoCF (2conv+50 words + attention1)	369k	0.4	2.0	1.3	2.0	5.1
BoCF (2conv+150 words + attention1)	376k	0.3	2.0	1.3	1.5	4.7
BoCF (2conv+200 words + attention1)	380k	0.3	2.0	1.2	1.5	5.0
BoCF (2conv+50 words + attention2)	15k	0.4	2.2	1.5	1.6	5.1
BoCF (2conv+150 words + attention2)	43k	0.3	2.0	1.2	1.4	4.8
BoCF (2conv+200 words + attention2)	63k	0.3	2.0	1.3	1.5	4.8
BoCF (SqueezeNet+150 words + no attention)	805k	0.4	2.2	1.4	1.7	5.3
BoCF (SqueezeNet+150 words + attention1)	892k	0.4	2.1	1.1	1.6	5.2
BoCF (SqueezeNet+150 words + attention2)	828k	0.4	2.0	1.6	1.4	5.0

illuminant and the estimated illuminant, defined as

$$\text{RAE}(I^{gt}, I^{Est}) = \cos^{-1}\left(\frac{I^{gt} I^{Est}}{\|I^{gt}\|\|I^{Est}\|}\right), \tag{15}$$

where I^{gt} is the ground truth illumination for a given image and I^{Est} is the estimated illumination. The mean of the worst 25% reflects how the model performs in the worst-case scenario and typically shows the largest differences between different methods..

V. EXPERIMENTAL RESULTS

In this section, we provide the experimental evaluation of the proposed method and its variants. In Subsection V-A, different topologies for the three blocks of BoCF are evaluated on the ColorChecker RECommended dataset and the effect of each block in our model is examined by reporting the results of the ablation studies. In Subsection V-B, we compare the performance of the proposed models with different state-of-the-art algorithms on the three datasets.

A. BoCF Performance Evaluation

We first evaluated the accuracy of the different variants of BoCF on ColorChecker RECommended dataset. Table I presents the comparative results for BoCF using different topologies in the three blocks. We evaluated the model using different numbers of convolution layers in the first block, different dictionary sizes in the second block (codewords), and with/without attention. In addition, in order to show the effect of using more convolutional layers, we used the pretrained model SqueezeNet [9] in the first block.

Table I shows that the dictionary size in the Bag-of-Features Pooling block significantly affects the overall performance of the model. Using a larger codebook results in higher risk of overfitting to the training data, while using a smaller codebook size restricts the model to only few codebook centers which can decrease the overall performance of the model. Thus, the choice of this hyperparameter is critical for our model. The findings in Table I confirm this effect and highlights the importance of this hyperparameter. By comparing the model performance using different dictionary sizes, we can see that a dictionary of size 150 yields the best compromise between the number of parameters and the overall performance.

Using three convolutional layers instead of two in the first block yields slightly better median errors and worse trimean errors. We note that using a pretrained model, squeezeNet, increases the number of parameters without improving the overall results. However, to keep the model as shallow as possible, we opt for the two convolution layers.

Table I shows that models equipped with an attention mechanism perform better than models without attention almost consistently across all error metrics. This is expected as attention mechanisms allow the model to focus on relevant parts only and, as a result, the model becomes more robust to noise and to inadequate features. The performance boost obtained by both attention variants is more highlighted in terms of the median and trimean errors compared to the non-attention variant.

By comparing the performance achieved by the two attention variants, we note that the first attention variant yields in a better performance in terms of worst 25% error rate, while the second variant yields better median and trimean error rates. It should also be remembered that the first variant applies attention over the feature map output of the first convolutional block. Thus, it dramatically increases the number of model parameters (over 20 times) compared to the second variant (doubling the number of parameters) which applies the attention over the histogram.

Figure 6 presents a visualization of the attention weights [50] for both attention variants. The heat maps demonstrate which regions of the image each model pays attention to so as to output a certain illumination. We note a large difference between the attention variants. The first attention variant tends to focus on regions with dense edges and sharp shapes, while the second model focuses on uniform regions to estimate the illumination. With the second variant of attention, the model learns to focus on homogeneous areas because attention is applied on top of a histogram. Thus, the model learns to focus on certain bins of this histogram. As each bin corresponds to the occurrence of one specific element of the dictionary, it is usually mapped to a homogeneous region of the image. With regards to the the first variant, one plausible explanation is that the models learns to focus on the edges. It is known that

Fig. 6. Attention mask visualization [50] for three samples from INTEL-TUT2, ColorChecker RECommended, and NUS8 datasets, respectively. The first column contains the input image. The second one illustrates the attention mask generated by the first attention variant overlaid on the input image. The last column contains the attention masks generated by the second variant of the attention overlaid on the input image. Gamma correction was applied for visualization.

TABLE II

RESULTS OF THE ABLATION STUDIES FOR THE BOCF OVER THE RECOMMENDED COLORCHECKER DATASET. BOCF IS THE BASIC BOCF COMPOSED OF THE THREE BLOCKS. IN BOCF-1, THE FEATURE EXTRACTION BLOCK IS REMOVED, WHILE IN BOCF-2 THE FULLY CONNECTED LAYER IN THE ESTIMATION BLOCK IS SUBSTITUTED WITH A LINEAR REGRESSION. IN BOCF-3, THE BOF POOLING LAYER IS REPLACED WITH A GLOBAL AVERAGE POOLING LAYER

Method	Best 25%	Mean	Med.	Tri.	Worst 25%
BoCF	0.3	2.1	1.5	1.6	5.1
BoCF-1	0.4	2.9	1.9	2.2	6.9
BoCF-2	0.5	2.4	1.7	1.7	5.7
BoCF-3	0.4	2.2	1.5	1.7	5.2

TABLE III

NUMBER OF PARAMETERS OF DIFFERENT CNN-BASED APPROACHES

Method	# parameters
Bianco [5]	154k
FC4 (SqueezeNet) [6]	1.9M
FC4 (AlexNet) [6]	3.8M
DS-Net [7]	17.3M
BoCF (2conv+150 words + no attention)	20k
BoCF (2conv+150 words + attention1)	376k
BoCF (2conv+150 words + attention2)	43k

the edges are an important feature in color constancy [25] and the model learns to extract and 'attend' this information to estimate the illumination.

Ablation studies

To examine the effect of each block in our proposed approach, we conduct ablation studies on the ColorChecker RECommended dataset. Table II reports the results of the basic BoCF approach, the results achieved by removing the feature extraction block, the results obtained by removing the estimation block, i.e., replacing the fully connected layer in the estimation block with a simple regression, and the results obtained by replacing the BoF block by a global average pooling layer [51]. We note that removing any block significantly decreases the overall performance of our models.

Comparing the model with and without the feature extraction block, we note a large drop in performance especially in terms of the worst 25% error rate, i.e., 1.8° drop compared to 0.6° drop when the estimation block is removed. We also note that BoF pooling performs better than global average pooling across all metrics except the median.

B. Comparisons Against State-of-the-art

We compared our BoCF approach with the state-of-the-art methods on ColorChecker RECommended, NUS-8, and INTEL-TUT2 datasets. Tables IV, V, and VI provide quantitative results for ColorChecker RECommended, NUS-8, and INTEL-TUT2 datasets, respectively. We provide results for the static methods Grey-World, White-Patch, Shades-of-Grey, and

TABLE IV
RESULTS OF BOCF APPROACH AND COMPARATIVE METHODS ON THE RECOMMENDED COLORCHECKER DATASET

Method	Type		Best 25%	Mean	Med.	Tri.	Worst 25%
	statistic-based	learning-based					
Grey-World [24]	✓	–	5.0	9.7	10	10	13.7
White-Patch [23]	✓	–	2.2	9.1	6.7	7.8	18.9
Shades-of-Gray [54]	✓	–	2.3	7.3	6.8	6.9	12.8
General-gray world [24]	✓	–	2.0	6.6	5.9	6.1	12.4
Pixel-based Gamut [30]	✓	–	1.7	6.0	4.4	4.9	12.9
Top-down [55]	✓	–	2.3	6.0	4.6	5.0	10.2
Spatial Correlations [56]	✓	–	1.9	5.7	4.8	5.1	10.9
Bottom-up [55]	✓	–	2.3	5.6	4.9	5.1	10.2
Edge-based Gamut [30]	✓	–	0.7	5.5	3.3	3.9	13.8
CC-GANs (Pix2Pix) [36]	–	✓	1.2	3.6	2.8	3.1	7.2
CC-GANs (CycleGAN) [36]	–	✓	0.7	3.4	2.6	2.8	7.3
CC-GANs (StarGAN) [36]	–	✓	1.7	5.7	4.9	5.2	10.5
FFCC (model Q) [33]	–	✓	0.3	2.0	1.1	1.4	5.1
Cheng et al. 2015 [57]	–	✓	0.4	2.4	1.7	1.7	5.9
DS-Net [7]	–	✓	0.3	1.9	1.1	1.4	4.8
CCC [52]	–	✓	0.3	2.0	1.2	1.4	4.8
Bianco CNN [5]	–	✓	0.8	2.6	2.0	2.1	4.0
FC4(SqueezeNet) [6]	–	✓	0.4	1.7	1.2	1.3	3.8
BoCF (2conv+150 words + no attention)	–	✓	0.3	2.1	1.5	1.6	5.1
BoCF (2conv+150 words + attention1)	–	✓	0.3	2.0	1.3	1.5	4.7
BoCF (2conv+150 words + attention2)	–	✓	0.3	2.0	1.2	1.4	4.8

TABLE V
RESULTS OF BOCF APPROACH AND BENCHMARK METHODS ON NUS-8 DATASET

Method	Type		Best 25%	Mean	Med.	Tri.	Worst 25%
	statistic-based	learning-based					
Grey-World [24]	✓	–	0.9	4.1	3.2	3.4	9.0
White-Patch [23]	✓	–	1.9	10.6	10.6	10.5	19.4
Shades-of-Gray [54]	✓	–	0.8	3.4	2.6	2.7	7.4
General-gray world [24]	✓	–	0.7	3.2	2.4	2.5	7.1
Pixel-based Gamut [30]	✓	–	2.5	7.7	6.7	6.9	14.0
Bright Pixels [58]	✓	–	0.7	3.2	2.4	2.6	7.0
Edge-based Gamut [30]	✓	–	2.4	8.4	7.0	7.4	16.1
Bayesian [11]	–	✓	0.8	3.7	2.7	2.9	8.2
Cheng et al. 2015 [57]	–	✓	0.6	2.9	2.0	2.2	6.6
DS-Net [7]	–	✓	0.5	2.2	1.5	1.7	6.1
CCC [52]	–	✓	0.5	2.4	1.5	1.7	5.9
Regression Tree [57]	–	✓	0.5	2.4	1.6	1.7	5.5
Bianco [5]	–	✓	0.3	2.6	2.0	2.1	3.9
FC4 (SqueezeNet) [6]	–	✓	0.5	2.2	1.5	1.7	5.2
FC4 (AlexNet) [6]	–	✓	0.5	2.1	1.6	1.7	4.8
BoCF (2conv+150 words + no attention)	–	✓	0.6	2.5	1.6	1.8	5.6
BoCF (2conv+150 words + attention1)	–	✓	0.5	2.3	1.4	1.7	5.2
BoCF (2conv+150 words + attention2)	–	✓	0.5	2.3	1.5	1.7	5.1

General Grey-World. The parameter values n, p, ρ are set as described in [25]. In addition, we compare against Pixel-based Gamut, Bright Pixels, Spatial Correlations, Bayesian Color Constancy [11], and six convolutional approaches: Deep Specialized Network for Illuminant Estimation (DS-Net) [7], Bianco CNN [5], Fast Fourier Color Constancy [33], Convolutional Color Constancy [52], Fully Convolutional Color Constancy With Confidence-Weighted Pooling (FC4) [6], and Color Constancy GANs (CC-GANs) [36]. The results for ColorChecker RECommended and NUS-8 datasets were taken from related papers [6], [36].

From RECommended ColorChecker and NUS-8 results in Tables IV and V, we note that the learning-based methods usually outperform the statistical-based methods across all error metrics. This can be explained by the fact that statistical approaches rely on some assumptions in their model. These assumptions can be violated in some testing samples which results in high error rates especially in terms of the worst 25% error.

TABLE VI
RESULTS OF BOCF APPROACH AND BENCHMARK METHODS ON INTEL-TUT2

Method	set	Best25%	Mean	Med.	Tri.	W.25%
Bianco CNN [5]	field	1.1	4.5	3.7	3.8	9.2
	non-field	1.8	6.2	5.3	5.5	12.4
C3AE [59]	field	1.6	4.4	4.0	4.2	7.9
fine-tuned	non-field	1.6	5.2	4.6	4.7	10.1
C3AE [59]	field	2.0	6.1	5.3	5.4	10.7
composite-loss	non-field	1.9	6.2	5.3	5.4	14.4
FC4 [6]	field	1.7	4.3	4.1	4.2	7.4
(SqueezeNet)	non-field	1.5	4.8	4.2	4.3	9.0
BoCF (150 w)	field	1.7	4.6	4.1	4.2	8.1
No attention	non-field	1.5	4.9	4.2	4.4	9.5
BoCF (150 w)	field	1.9	4.5	4.1	4.2	7.3
attention1	non-field	1.5	4.9	4.2	4.3	9.0
BoCF (150 w)	field	1.7	4.4	4.1	4.2	7.5
attention2	non-field	1.5	4.9	4.3	4.4	9.1

Table IV shows that the proposed method BoCF variants achieve competitive results on RECommended ColorChecker dataset. The only models performing slightly better than BoCF

TABLE VII

RESULTS OF BOCF APPROACH AND COMPARATIVE METHODS ON THE RECOMMENDED COLORCHECKER DATASET USING REPRODUCTION ANGULAR ERROR METRIC

Method	Type		Mean	Med.	Tri.
	statistic-based	learning-based			
Grey-World [24]	✓	–	10.1	7.5	8.3
White-Patch [23]	✓	–	9.7	7.4	8.2
Shades-of-Gray [54]	✓	–	6.9	3.9	8.2
General-gray world [24]	✓	–	6.0	3.9	4.3
Pixel-based Gamut [30]	✓	–	6.9	5.2	5.7
Edge-based Gamut [30]	✓	–	6.9	4.6	5.2
Bianco CNN [5]	–	✓	5.7	4.7	5.0
FFCC (model Q) [33]	–	✓	2.5	1.4	1.8
BoCF(2conv+150 words + no attention)	–	✓	2.6	1.8	2.0
BoCF(2conv+150 words + attention1)	–	✓	2.5	1.6	1.8
BoCF(2conv+150 words + attention2)	–	✓	2.3	1.5	1.8

are FC4(SqueezeNet) and DS-Net. By comparing the number of parameters required by each model given in Table III, we see that BoCF achieves very competitive results, while using less than 1% of the parameters of FC4(SqueezeNet) and less than 0.1% of the parameters of DS-Net.

Compared to Bianco CNN, we note that our model performs better across all error metrics except for the worst 25% error metric. Bianco CNN operates on patches instead of the full image directly and this makes it more robust but, at the same time, it increases its time complexity as the network has to estimate many local estimates before outputting the global one.

Results for NUS-8 dataset are similar to their counter parts on ColorChecker RECommended, as illustrated in Table V. Our models achieve comparable results with FC4 and overall better results compared to DS-Net across all error metrics. Bianco CNN outperforms all the other CNN-based methods. As discussed earlier, this can likely be explained by the fact that Bianco operates on patches while BoCF and FC4 produce global estimates directly.

Table VI reports the comparative results achieved on INTEL-TUT2 dataset. We note that all the error rates are high as this is an extreme testing scenario. The models are trained and validated using only one type of scene (*field2* set) acquired by one camera model (Canon) and then evaluated over different scene types and different camera models not seen during the training as described in Section IV-C. The proposed BoCF model achieves better overall performance compared to Bianco CNN on the *non-field* set, on both sets compared to Color Constancy Convolutional AutoEncoder (C3AE) methods and competitive results compared to FC4. The model is robust in transfer learning because it is invariant to the spatial location of the extracted feature (this is the strong point of the BoF). So, by having a representation that is "by construction" invariant to spatial changes makes the network less sensitive to spatial changes, and as a result, more robust.

By comparing the performance achieved by BoCF with and without attention, we note that both the attention mechanisms proposed in this paper significantly boost the performance of our model for all datasets. It should also be mentioned that despite requiring much less parameters, the second variant of our attention model, where the attention is applied over the histogram representation, performs slightly better than the first variant, where the attention is applied over the feature extraction block.

We also compare the performance of the proposed method based on Reproduction Angular Error [53] with other methods, which have been evaluated using this metric in the literature. The comparison are shown in Table 7. We note that our models outperform Bianco CNN across all metrics and the second variant of attention outperforms FFCC model in terms of mean error, while achieving similar performance in terms of trimean.

VI. DISCUSSION

When comparing our approach to the competing methods, it must be pointed out that our approach can be linked to many static-based approaches. In Grey-World [24], one takes the average of the RGB channels of the image. In the proposed method, this corresponds to using the identity as a feature extractor and using equal weights in the estimation block. This way all the histogram bins will contribute equally in the estimation. White-Patch [23] takes the maximum across the color channels, which corresponds to giving a high weight to the histogram bin with the highest intensity and giving zero weights to the rest. Grey-edge and its variants [25] correspond to using the first and second order derivatives as a feature extractor. Thus, BoCF approach can be interpreted as a learning-based extension of these statistical based approaches. Instead of using the image directly, we allow the model to learn a suitable non-linear transformation of the original image, through the feature extraction block, and instead of imposing a prior assumption on the contribution of each feature in the estimation, we allow the model to learn the mapping dynamically using the training data. BoF provides the link between these two tasks as it allows us to learn a dictionary and to output a histogram representation of the transformed image.

It is interesting to note that the attention variants in our approach can be tightly linked to the confidence maps in FC4 [6]. In FC4, confidence scores are assigned to each patch of the image and a final estimate is generated by a weighted sum of the scores and their corresponding local estimates. This way the network learns to select which features contribute to the estimation and which parts should be discarded. Similarly, attention mechanism learns to dynamically pay attention to the

parts encoding the illumination information and discarding the rest.

VII. Conclusion

In this paper, we proposed a novel color constancy method called BoCF, which is composed of three blocks. In first block, called feature extraction, we employ convolutional layers to extract relevant features from the input image. In the second block, we apply Bag-of-Features Pooling to learn a codebook and output a histogram. The latter is fed into the last block, the estimation block, where the final illumination is estimated. This end-to-end model is evaluated and compared with prior works over three datasets: ColorChecker RECommended, NUS-8, and INTEL-TUT2. BoCF was able to achieve competitive results compared to state-of-the-art methods while reducing the number of parameters up to 95%. In this paper, we also discussed links between the proposed method and statistic-based methods and we showed how the proposed approach can be interpreted as a supervised extension of these approaches and can act as a generic framework for expressing existing approaches as well as developing new powerful methods.

In addition, we proposed combining the Bag-of-Features Pooling with two novel attention mechanisms. In the first variant, we apply attention over the nonlinear transform of the image after the feature extraction block. In the second extension, we apply attention over the histogram representation of the Bag-of-Features Pooling. These extensions are shown to improve the overall performance of our model.

In future work, extensions of the proposed approach could include exploring regularization techniques to ensure diversity in the learned dictionary and improve the extension capability of the model.

References

[1] M. Ebner, *Color Constancy*, 1st ed. Hoboken, NJ, USA: Wiley, 2007.

[2] J. J. M. Granzier, E. Brenner, and J. B. J. Smeets, "Can illumination estimates provide the basis for color constancy?" *J. Vis.*, vol. 9, no. 3, p. 18, 2009.

[3] K. Barnard, "Practical colour constancy," Ph.D. dissertation, School Comput., Simon Fraser Univ., Burnaby, BC, Canada, 1999.

[4] A. Gijsenij, T. Gevers, and J. van de Weijer, "Computational color constancy: Survey and experiments," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2475–2489, Sep. 2011.

[5] S. Bianco, C. Cusano, and R. Schettini, "Color constancy using CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2015, pp. 81–89.

[6] Y. Hu, B. Wang, and S. Lin, "FC^4: Fully convolutional color constancy with confidence-weighted pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4085–4094.

[7] W. Shi, C. C. Loy, and X. Tang, "Deep specialized network for illuminant estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 371–387.

[8] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, and J.-F. Lalonde, "Deep outdoor illumination estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2373–2382.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[10] C. Aytekin, J. Nikkanen, and M. Gabbouj, "A data set for camera-independent color constancy," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 530–544, Feb. 2018.

[11] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian color constancy revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[12] G. Hemrit *et al.*, "Rehabilitating the colorchecker dataset for illuminant estimation," in *Proc. Color Imag. Conf.*, 2018, pp. 350–353.

[13] D. Cheng, D. K. Prasad, and M. S. Brown, "Illuminant estimation for color constancy: Why spatial-domain methods work and the role of the color distribution," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 31, no. 5, pp. 1049–1058, 2014.

[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: https://arxiv.org/abs/1602.07360

[15] H. C. Lee, "Method for computing the scene-illuminant chromaticity from specular highlights," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 3, no. 10, pp. 1694–1699, 1986.

[16] N. Passalis and A. Tefas, "Neural bag-of-features learning," *Pattern Recognit.*, vol. 64, pp. 277–294, Apr. 2017.

[17] N. Passalis and A. Tefas, "Learning bag-of-features pooling for deep convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5755–5763.

[18] F.-F. Li and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 524–531.

[19] G. Qiu, "Indexing chromatic and achromatic patterns for content-based colour image retrieval," *Pattern Recognit.*, vol. 35, no. 8, pp. 1675–1686, Aug. 2002.

[20] A. Iosifidis, A. Tefas, and I. Pitas, "Discriminant bag of words based representation for human action recognition," *Pattern Recognit. Lett.*, vol. 49, pp. 185–192, Nov. 2014.

[21] A. Iosifidis, A. Tefas, and I. Pitas, "Multidimensional sequence classification based on fuzzy distances and discriminant analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2564–2575, Nov. 2013.

[22] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proc. ACM Int. Conf. Image Video Retr.*, 2007, pp. 494–501.

[23] A. Rizzi, C. Gatta, and D. Marini, "Color correction between gray world and white patch," *Int. Soc. Opt. Eng.*, vol. 4662, pp. 367–375, May 2002.

[24] J. Cepeda-Negrete and R. E. Sanchez-Yanez, "Gray-world assumption on perceptual color spaces," in *Proc. Pacific-Rim Symp. Image Video Technol.* Berlin, Germany: Springer, 2013, pp. 493–504.

[25] J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-based color constancy," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2207–2214, Sep. 2007.

[26] X.-S. Zhang, S.-B. Gao, R.-X. Li, X.-Y. Du, C.-Y. Li, and Y.-J. Li, "A retinal mechanism inspired color constancy model," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1219–1232, Mar. 2016.

[27] R. Tan, K. Nishino, and K. Ikeuchi, "Color constancy through inverse-intensity chromaticity space," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 21, no. 3, pp. 321–334, 2004.

[28] J. Vazquez-Corral, M. Vanrell, R. Baldrich, and F. Tous, "Color constancy by category correlation," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1997–2007, Apr. 2012.

[29] J. Fröhlich, A. Schilling, and B. Eberhardt, "Gamut mapping for digital cinema," in *Proc. SMPTE Annu. Tech. Conf. Exhib.*, 2013, pp. 1–11.

[30] G. Finlayson and S. Hordley, "Improving gamut mapping color constancy," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1774–1783, Oct. 2000.

[31] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Automatic color constancy algorithm selection and combination," *Pattern Recognit.*, vol. 43, no. 3, pp. 695–705, Mar. 2010.

[32] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini, "Improving color constancy using indoor–outdoor image classification," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2381–2392, Dec. 2008.

[33] J. T. Barron and Y.-T. Tsai, "Fast Fourier color constancy," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 886–894.

[34] R. Lu, A. Gijsenij, T. Gevers, V. Nedovic, D. Xu, and J.-M. Geusebroek, "Color constancy using 3D scene geometry," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1749–1756.

[35] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[36] P. Das, A. S. Baslamisli, Y. Liu, S. Karaoglu, and T. Gevers, "Color constancy by GANs: An experimental survey," 2018, *arXiv:1812.03085*. [Online]. Available: https://arxiv.org/abs/1812.03085

[37] N. Passalis, A. Tsantekidis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Time-series classification using neural bag-of-features," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2017, pp. 301–305.

[38] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473
[39] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[40] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 551–561.
[41] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," 2017, *arXiv:1703.03906*. [Online]. Available: https://arxiv.org/abs/1703.03906
[42] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
[43] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.
[44] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[45] S. Gao, M. Zhang, C. Li, and Y. Li, "Improving color constancy by discounting the variation of camera spectral sensitivity," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 34, no. 8, pp. 1448–1462, 2017.
[46] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io
[47] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement.*, 2016, pp. 265–283.
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[49] S. D. Hordley and G. D. Finlayson, "Re-evaluating colour constancy algorithms," in *Proc. Int. Conf. Pattern Recognit.*, 2004, pp. 76–79.
[50] R. Kotikalapudi. (2017). *keras-vis*. [Online]. Available: https://github.com/raghakot/keras-vis
[51] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. Int. Conf. Learn. Represent.*, 2014.
[52] J. T. Barron, "Convolutional color constancy," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 379–387.
[53] G. D. Finlayson, R. Zakizadeh, and A. Gijsenij, "The reproduction angular error for evaluating the performance of illuminant estimation algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1482–1488, Jul. 2017.
[54] G. Finlayson and E. Trezzi, "Shades of gray and colour constancy," in *Proc. Color Imag. Conf.*, 2004, pp. 37–41.
[55] J. van de Weijer, C. Schmid, and J. Verbeek, "Using high-level visual information for color constancy," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[56] A. Chakrabarti, K. Hirakawa, and T. Zickler, "Color constancy with spatio-spectral statistics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1509–1519, Aug. 2012.
[57] D. Cheng, B. Price, S. Cohen, and M. S. Brown, "Effective learning-based illuminant estimation using simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1000–1008.
[58] H. R. V. Joze, M. S. Drew, G. D. Finlayson, and P. A. T. Rey, "The role of bright pixels in illumination estimation," in *Proc. Color Imag. Conf.*, 2012, pp. 41–46.
[59] F. Laakom, J. Raitoharju, A. Iosifidis, J. Nikkanen, and M. Gabbouj, "Color constancy convolutional autoencoder," in *Proc. Symp. Comput. Intell.*, Dec. 2019, pp. 1085–1090.

Firas Laakom received the Engineering degree from Tunisia Polytechnic School (TPS) in 2018. He is currently pursuing the Ph.D. degree with Tampere University, Finland. His research interests include deep learning, computer vision, and computational intelligence.

Nikolaos Passalis is currently a Postdoctoral Researcher with Tampere University, Finland. He has coauthored more than 45 journal articles and conference papers and contributed one chapter to one edited book. His research interests include machine learning, information retrieval, and computational intelligence.

Jenni Raitoharju (Member, IEEE) received the Ph.D. degree from Tampere University of Technology, Finland, in 2017. Since 2017, she has worked as a Post-Doctoral Research Fellow with the Faculty of Information Technology and Communication Sciences, Tampere University, Finland. In September 2019, she started working as a Senior Research Scientist with the Finnish Environment Institute, Jyväskylä, Finland, after receiving Academy of Finland Post-Doctoral Researcher funding from 2019 to 2022. She has coauthored 15 journal articles and 27 conference papers. Her research interests include machine learning and pattern recognition methods along with applications in biomonitoring and autonomous systems. She is the Chair of the Young Academy Finland 2019–2020.

Jarno Nikkanen received the M.Sc. degree in signal processing and Dr.Sc.Tech. degree in software systems from the Tampere University of Technology in 2001 and 2013, respectively. He has 20 years of industry experience in digital imaging topics, including Nokia Corporation where he developed and productized many digital camera algorithms, Intel Corporation where he worked as the Intel Principal Engineer and Imaging Technology Architect, and Xiaomi Technologies where he is currently working as the Head of Xiaomi Finland camera Research and Development and the Leader of Xiaomi's Tampere site. He holds international patents for over 20 digital camera related inventions and multiple additional patents pending.

Anastasios Tefas (Member, IEEE) is currently an Associate Professor with the Department of Informatics, Aristotle University of Thessaloniki. He has coauthored 120 journal articles, 235 papers in international conferences and contributed eight chapters to edited books in his area of expertise. Over 6000 citations have been recorded to his publications and his H-index is 37 according to Google scholar. His current research interests include computational intelligence, deep learning, digital signal and image analysis and retrieval, and computer vision.

Alexandros Iosifidis (Senior Member, IEEE) is currently an Associate Professor with Aarhus University, Denmark. He has contributed in more than twenty Research and Development projects financed by EU, Greek, Finnish, and Danish funding agencies and companies. He has coauthored 71 articles in international journals and 88 papers in international conferences proposing novel machine learning techniques/methods and their application in a variety of problems. His current research interests include machine learning solutions for problems coming from signal processing, computer vision, and financial modeling.

Moncef Gabbouj (Fellow, IEEE) received the M.S. and Ph.D. degrees in EE from Purdue University, in 1986 and 1989, respectively. He is currently a Professor of signal processing with the Department of Computing Sciences, Tampere University, Finland. His research interests include big data analytics, multimedia analysis, artificial intelligence, machine learning, pattern recognition, nonlinear signal processing, video processing, and coding. He is a member of the Academia Europaea and the Finnish Academy of Science and Letters.