COCOA: Combining Color Constancy Algorithms for Images and Videos

Simone Zini, Marco Buzzelli, Simone Bianco, and Raimondo Schettini

***Abstract*—We present an efficient combination strategy for color constancy algorithms. We define a compact neural network architecture to process and combine the illuminant estimations of individual algorithms, that may be based on different assumptions over the input scene content. Our solution can be specialized to the image domain, thus expecting a single frame input, and to the video domain, exploiting a Long Short-Term Memory module (LSTM) to handle varying-length sequences. To prove the effectiveness of our combining method we limit ourselves to combine only learning-free color constancy algorithms based on simple image statistics. We experiment on the standard Shi-Gehler and NUS datasets for still images, and on the recent Burst Color Constancy dataset for videos. Experimental results show that our method outperforms other combination strategies, and reaches an illuminant estimation accuracy comparable to more sophisticated and computationally-demanding solutions when the standard dataset split is used. Furthermore, our solution is proven to be effective even when the number of training instances available is reduced. As a further analysis, we assess the individual contribution of each underlying method towards the final illuminant estimation.**

***Index Terms*—Algorithms combination, color constancy, CNN, deep learning, illuminant estimation, information fusion, LSTM.**

I. Introduction

COMPUTATIONAL color constancy aims at reducing the chromatic dominant in a digital image, originated from the light source that illuminates the scene. This goal is typically pursued through the development of an algorithm for illuminant estimation. The research community has been tackling the problem of computational color constancy for several years, designing a disparate set of approaches, which range from handcrafted methods based on low-level image statistics, to data-driven methods based on middle-to-high level analysis. Each individual approach necessarily exploits a specific set of biases and rationales. Due to the ill-posed nature of the problem, in fact, color constancy is not mathematically solvable without relying on additional assumptions on the imaged content. For example, the edge-based color constancy framework by van de Weijer et al. [1] describes with a unified formulation several low-statistics algorithms that are based on different assumptions: the gray-world hypothesis, which assumes that the average reflectance in a scene is achromatic, the white patch hypothesis, based on the assumption that the reflectance achieved for each of the color channels is equal, or the gray-edge hypothesis, according to which the average of the reflectance differences in a scene is achromatic. More recent data-driven methods, such as deep learning solutions, implicitly operate higher-level abstractions, by both exploiting statistical biases in the training data, as well as associations with the semantics of the image content. However the most recent and most performing methods are increasingly computationally and memory demanding, and require large dataset to be properly trained. Many methods for color constancy operate under the assumption that a single illuminant is present in the scene, assumption which is also adopted within this paper. We refer the reader to Hussain et al. [2], for an overview of multiple-illuminant color constancy methods, which are classified as based on local light estimation, pixel detection, convolutional neural networks, or biologically-inspired, and for a solution that exploits image texture to select pixels with sufficient color variation to be used for image color correction.

Since different color constancy methods often rely on different assumptions, they can be expected to provide different and uncorrelated outputs, and to consequently perform better on different types of input. The illuminants estimated by these methods, being influenced by the underlying assumptions, can then be considered as image-describing features, and thus properly combined through a fusion strategy for improved color constancy. This approach has been successfully adopted in a wide range of domains, from change detection algorithms for background segmentation [3], to saliency estimation methods [4], to color constancy itself [5], [6], [7]. One of the main drawbacks of the fusion approach is often represented by the inference time, as it requires running multiple independent algorithms on the same input, and subsequently combining the results with a sufficiently-advanced fusion strategy. This aspect becomes particularly problematic in a video-oriented domain, where time is considered critical. In such a scenario, therefore, it is fundamental to 1) select efficient input methods, possibly sharing common processing steps to reduce the computational overhead, and 2) develop an efficient combination strategy. Barron et al. [8] report a threshold of 30 frames per seconds (FPS) to consider an algorithm viable for application in the camera viewfinder stream. The same threshold is also commonly accepted in other fields, such as responsive systems for assisted driving [9]. Even in an off-line color constancy setup, where live-feedback is not

Manuscript received 2 March 2022; revised 5 July 2022; accepted 15 August 2022. Date of publication 2 September 2022; date of current version 21 September 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Matthias Hullin. *(Corresponding author: Simone Bianco.)*

The authors are with the Department of Informatics, Systems and Comunnication, University of Milano-Bicocca, 20126 Milano, Italy (e-mail: simone.zini@unimib.it; marco.buzzelli@unimib.it; simone.bianco@unimib.it; raimondo.schettini@unimib.it).

Source code available at: https://github.com/TheZino/COCOA.
Digital Object Identifier 10.1109/TCI.2022.3203889

required, a fast computation is still critical for the processing of long video sequences.

In this paper, assuming to have a set of input color constancy methods, not necessarily the most effective ones, we design a very efficient late-fusion combination strategy that is able to reach an accuracy close to the best algorithms in the state of the art, keeping at the same time the computational burden also suitable for the real time video domain. We apply our single-frame lightweight combination strategy to a selection of methods based on simple image statistics [1], proving to be effective even when an extremely limited amount of training data is available. We outperform other combination strategies on a standard dataset for single-frame color constancy, and reach an illuminant estimation accuracy comparable to more sophisticated solutions.

We also present an extension of our fusion strategy that exploits a Long Short-Term Memory (LSTM) module to handle varying-length video sequences. Experiments on the recent Burst Color Constancy dataset (BCC) [10] show that: i) exploiting the temporal component after the combination gives better results than exploiting it before the combination; ii) the proposed method outperforms other strategies that can be implemented to exploit the temporal component; iii) the proposed method is able to reach an illuminant estimation accuracy on video sequences comparable to more sophisticated and computationally-demanding solutions specifically designed for video applications.

We also evaluate our solution in terms of inference time, showing how the combination represents a negligible overhead on the computational time required by the combined algorithms. By exploiting and optimizing the redundancies of the underlying set of input methods, we are able to reach real time performance at 31 frames per seconds. Finally, we conduct a series of experiments aimed at analyzing the behavior of the proposed combining method, and at assessing the individual contribution of each underlying method towards the final illuminant estimation.

II. Related Works

A. Single-Frame Combinational Illuminant Estimation Methods

Combinational illuminant estimation methods give an estimate of the scene illuminant by combining the estimates given by a set of input methods. Combinational illuminant estimation methods have been reviewed in [11], where they have been categorized into two main classes on the basis of the information they use as input: direct combination methods (DC) provide their final estimate as a combination of the estimates given by the input methods to be combined; guided combination methods (GC) exploit additional information extracted from the input image, in terms of semantic class or features, together with the the estimates given by the input methods to be combined. Direct combination methods have been further grouped into supervised combination (SC) and unsupervised combination (UC) methods: the former ones have a training phase to learn how to combine the estimates given by the input methods, while the latter ones directly combine them without any training phase.

Concerning the DC methods, Cardei and Funt proposed two combining methods [5]: Simple Committee, belonging to the UC methods since the combination is performed by simply averaging the estimates of the combined algoriths, and LMS Committee, belonging to the SC methods where the combination weights are learned in a Least Mean Squares optimization.

Bianco et al. [6] proposed a set of different DC-UC methods by exploiting the spatial positions of the estimations to be combined. Considering the estimates as points in the space, Nearest-X averages the estimates of the X algorithms that are closest between each other. The Nearest-X% combination averages all the estimates for which the distance between any pair of them is below $(100 + X)\%$ of that between the two closest ones. The No-N-Max method instead averages the estimates excluding the N estimates having the highest distance from the other estimates. The last method they propose is the Median combinational strategy that selects the estimate having the smallest total distance from all the others.

Li et al. [7] proposed two DC-SC methods: the first uses an Extreme Learning Machine to perform the combination, while the second exploits a Support Vector Regression.

GC methods exploit additional information extracted from the image to drive the combination: in [12] each image is described by a set of low-level features related to color, texture, and edge distribution and exploits tree-based image classifier trained on indoor, outdoor, close-up classes; [13] uses general-purpose features and problem dependent low-level features without the need of a proxy constituted by semantic classes; a similar approach is used in [14], that exploits texture and contrast summarized in terms of the Weibull parameterization; [15] uses high-level visual information to improve illuminant estimation by modelling the image as a mixture of semantic classes, such as sky, grass, road, and building; [16] uses rough 3D scene geometry to model an image in terms of different geometrical regions and depth layers.

Given the success of the above combining methods Li et al. [17] proposed a multi-cue method that combines the information provided by different cues, e.g. properties of the low-level RGB color distribution, mid-level initial illuminant estimates provided by subordinate method, and high-level knowledge of scene content, within the framework of a tree-structured group joint sparse representation.

Subhashdas et al. [18], [19] propose a hybrid multi-class dynamic weight model with an ensemble of classifiers: their method classifies images into several groups and uses a distinct dynamic weight generation model (DWM) for each group. The DWM generates dynamic weight using an image feature that has a correlation with the capability of the input algorithms used for combination.

B. Video Illuminant Estimation Methods

Although frame-based illuminant estimation methods can be applied also to videos and/or image sequences on a per-frame basis, there are only a few methods actually able to exploit the temporal component to produce a more robust illuminant estimate.

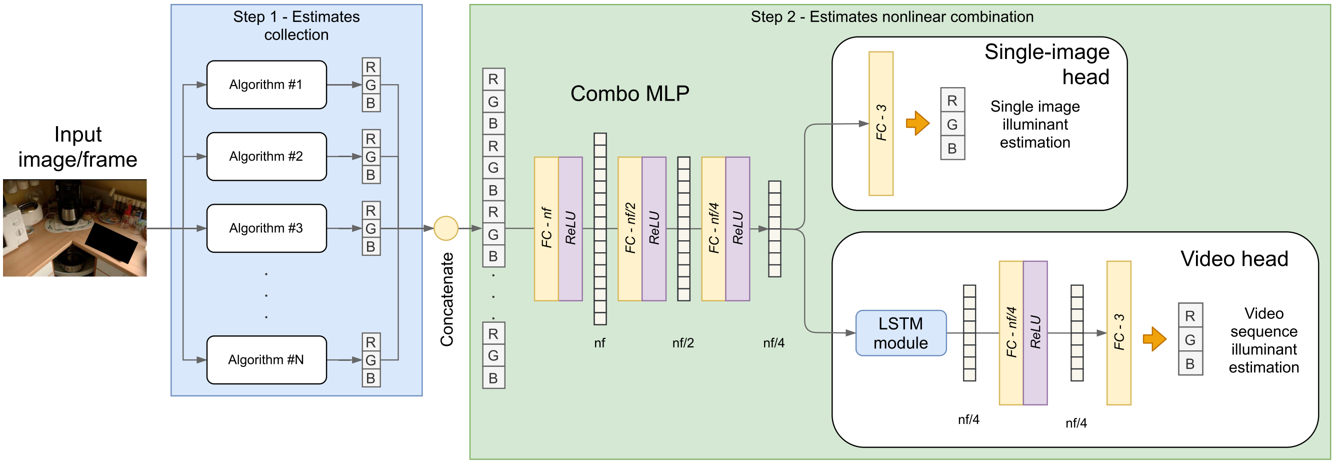


Fig. 1. Combination framework for illuminant estimation combination. The framework is composed of two different steps: the first one corresponds to the collection of the statistics-based approaches estimations, the second one corresponds to the actual combination of the estimations previously collected. As can be seen, the Single-Image model and the Video model shares the same architecture for the first combination part: the two models differs for the different heads. In the Single-Image case, the head is made of only one linear layer, used to map the $nf/4$ features to the output dimensionality. For the Video model, the $nf/4$ features are further processed by a LSTM to exploit the temporal nature of the video sequence. The details of the video sequence processing are shown in Fig. 2.

Yang et al. [20] extract illuminant color from two distinct frames of the same scene exploiting highlights on specular surfaces. Prinet et al. [21] propose a probabilistic and more robust version of [20].

Wang et al. [22] propose a multi-frame illuminant estimation method by clustering illuminant estimate coming from a standard method on each frame into a number of video shots and then exploit a summary statistics to provide a global estimate for the whole shot.

More recently, Barron et al. [8] extended their single frame method to work on image sequences by building a smoothing model inspired by Kalman filter in order to smooth wrong predictions that may happen on individual frames.

The work of Quian et al. [23] is the first to actually exploit the information available in the input sequence. They propose an end-to-end trainable recurrent color constancy network that exploits AlexNet features and a Long Short Term Memory (LSTM) recurrent neural network to process sequential input frames. Their method has been then improved [10] by exploiting a more powerful backbone network for the semantic feature extraction, and using a 2D LSTM that provides more effective spatial recurrent information.

III. Proposed Method

We propose a framework for the non-linear combination of illuminant estimations, using a small neural network composed of few hidden layers in a multilayer perceptron (MLP) architecture. The general idea is to exploit the different assumptions related to different illuminant estimation algorithms. We designed two variants of this model: a single-image illuminant estimation version, and a video estimation version operating on multiple input frames. In this section we are going to present the framework for both configurations, with the respective architectures and the objective function adopted for training.

A. Single-Image Model

The proposed framework for illuminant estimations combination is illustrated in Fig. 1. The procedure is divided into two steps: the first step consists in performing the initial illuminant estimation using a given set of algorithms, in order to collect the different estimations to be combined. The second step corresponds to using our multilayer perceptron, called COCOA, to obtain the corresponding non-linear combination of the input estimations.

The COCOA network is a multilayer perceptron model made of four linear layers which uses Rectifying Linear Unit (ReLU) activation functions. The structure of COCOA is represented in Fig. 1. As can be seen from Fig. 1 the number of perceptrons per layer is defined as a function of the number of perceptrons in the first layer. In our configuration we adopted $\text{nf} = 256$, obtaining a four-layer model with respectively 256, 128, 64, and 3 perceptrons.

Given a set of algorithms for combination, we train the COCOA model by giving as input the concatenation of the estimations, in normalized RGB space, and compare the output combination with the ground truth. The number of algorithms used to obtain the starting estimations determine the dimensionality of the first layer of COCOA.

B. Video Model

We present a variant of the proposed model, specifically designed for the processing of video sequences.

In this scenario, for each frame in a given sequence, our model takes as input a set of estimations, performed with a set of input illuminant estimation methods, and extracts a vector of $\text{nf}/4$ features. This part of the model corresponds to the first three layers of the single-image illuminant estimation model. The resulting features are are then processed by a Long Short-Term Memory module (LSTM).

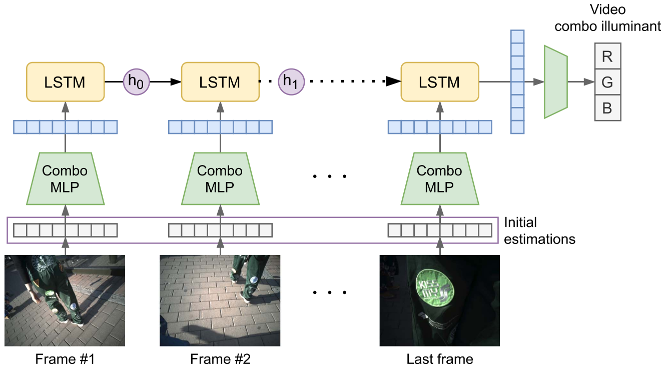


Fig. 2. Combination of the illuminant estimations between frames of a video sequence. For each frame the 6 estimations are first processed by the Combo MLP component, then are given in input to the LSTM module. Finally the processed features are sent to the last two layers, giving in output the final estimation.

For each frame, the LSTM module takes in input the representation given by the MLP and generates a new set of features, representing the frame sequence until the last processed frame. For each frame of the sequence, the MLP feature extraction step with the LSTM module temporal processing is repeated, using as input the estimations of the input methods corresponding to the new frame, and the hidden state coming from the previous step of the LSTM module (with exception for the first frame). The final results coming from the processing of each frame is eventually passed to a final group of two fully connected layers, which outputs the estimation for the entire sequence. The video estimations combination process is depicted in Fig. 2. As can be seen from Fig. 1, the model for the video estimation combination is an extension of the original single-image model presented in Section III-A. Instead of having a final layer which maps the nf/4 representation to the output dimensionality 3, we have the new components which handle the multi-frame nature of the video sequence.

The LSTM is initialized with starting hidden state and starting cell state at zero values, with hidden state dimension equal to nf/4. For the last two layers, as can be seen in Fig. 1, the number of output features for the two fully connect layers is respectively nf/4 (64 in our configuration using nf = 256) and 3.

C. Loss Function

We train the two models by minimizing the recovery angular error (expressed in degrees) between the output of COCOA and the ground truth illuminant associated to the image or to the video sequence. In the case of the single-image model, we have a ground truth illuminant for each image in the dataset, while for the video case scenario, for each video sequence in the dataset we have a single ground truth illuminant triplet for the entire video sequence. This is determined by the chosen datasets for experimentation, as illustrated in Section IV-B.

The recovery angular error, which quantifies the illuminant estimation error, is represented by the angle between the vector given by the target illuminant triplet $\rho^{gt} = (R^{gt}, G^{gt}, B^{gt})$ and the one corresponding to the result of the combination $\rho^E = (R^E, G^E, B^E)$. Given two illuminants ρ^E and ρ^{gt}, the recovery angular error can be calculated as:

$$\theta = \arccos\left(\frac{(\rho^E \cdot \rho^{gt})}{||\rho^E|| \cdot ||\rho^{gt}||}\right) \tag{1}$$

IV. Experimental Setup

A. Training Setup

The COCOA architecture is written in Pytorch 1.7.0 and trained on an NVIDIA Titan V with 12 GB of memory. Training was performed using the Adam [24] optimizer; for the single image scenario we selected a starting learning rate of 0.003 and weight decay of 1e-5, while for the multi-frame training we used a starting learning rate of 1e-4 and weight decay of 1e-5. Both models were trained for a total amount of 3000 epochs.

B. Datasets

To train and evaluate the performance of the COCOA model we adopted different setups and datasets for the image and video tasks. For single-image illuminant estimation we used the 569 images of the Shi-Gehler reprocessed dataset [25], [26], while for the video illuminant estimation task we used the 600 sequences of the Burst Color Constancy dataset (BCC) from Qian et al. [10].

For single-image illuminant estimation on the Shi-Gehler dataset we adopted the original three-fold cross validation division, as done previously by [8], [27], and we performed validation by randomly selecting the 20% of the training images for each fold. For the video dataset we used the original dataset division provided by the authors, for training and test. To validate the model we randomly selected 20% of the video sequences from the training set.

C. Combined Input Methods

For each image or frame, we collect six different illuminant estimations from different statistics-based algorithms:

- Shades of Gray (SoG)
- General Gray World (gGW)
- Gray Edge 1st order (GE1)
- Gray Edge 2nd order (GE2)
- Gray World (GW)
- White point (WP)

This particular selection aims at creating an overall illuminant estimation pipeline that is also practical, i.e. by relying on simple input methods, its computational complexity remains low and suitable for a real-time application, as shown in Section V-D. To perform the illuminant estimation using those models, we adopted the framework from van de Weijer et al. [1], which offers a single equation to perform the illuminant estimation corresponding to different assumptions over the images. The general hypothesis is described as:

$$\left(\int \left|\frac{\partial^n f^\sigma(x)}{\partial x^n}\right|^p dx\right)^{1/p} = ke^{n,p,\sigma} \tag{2}$$

where n identifies the derivative order, σ is the standard deviation for a Gaussian filter, implemented with a convolutional kernel

TABLE I
PARAMETERS FOR EACH ILLUMINANT ESTIMATION ALGORITHM. THE FREE PARAMETERS THAT CAN BE CHANGED WITHOUT SWITCHING TO A DIFFERENT METHOD ARE HIGHLIGHTED IN BOLDFACE

	COCOA			COCOA-fast		
	n	p	σ	n	p	σ
Shades Of Gray (SoG)	0	**4**	0	0	**1**	0
Gray World (GW)	0	1	0	0	1	0
Gray Edge 1st order (GE1)	1	**1**	**6**	1	**1**	**1**
Gray Edge 2nd order (GE2)	2	**1**	**1**	2	**1**	**1**
general Gray World (gGW)	0	**9**	**9**	0	**1**	**1**
White Patch (WP)	0	∞	0	0	∞	0

whose size is directly proportional to σ itself, and p is the order of the Minkowski norm. These parameters can be controlled to specialize the behavior of the six algorithms listed above. More precisely, we define two configurations, as reported in Table I, aimed respectively at optimizing estimation accuracy or speed. The first configuration (COCOA) uses values reported from the state of the art [28]. Although these parameters are arbitrary, they are representative of a specific use case: a setup that is potentially optimal in terms of estimation accuracy. The second configuration (COCOA-fast) is obtained by reducing the free parameters, focusing on efficiency more than effectiveness. Here the specific choice of values is driven by the following motivations: 1) to speed up the computation by sharing common processing steps among multiple methods, 2) to further speed up the computation by only exploiting a small convolutional kernel for the Gaussian filter, and 3) to avoid relying on arbitrary parameters that are potentially optimized to a specific dataset. As it can be observed from the table, this second set of parameters has the Shades of Gray algorithm collapse into a Gray World, thus reducing the effective total number of input methods from six to five.

The camera black level is subtracted from all images, and these are subsequently rescaled to have their maximum side be 256 pixels long. After this pre-processing, we eventually feed each image into each one of the algorithms, obtaining a total amount of six estimations per image. These six estimations are first normalized and then concatenated and used as input for the COCOA model. We conducted a series of preliminary experiments to define the most appropriate normalization strategy, including L2 normalization, green channel normalization, and conversion to various chromaticity representations. The final configuration, adopted throughout all our experiments, relies on green-channel normalization. The final output of the network consists of an RGB triplet corresponding to the non-linear combination of the input estimates.

V. EXPERIMENTAL RESULTS

A. *Combinational Single-Image Illuminant Estimation*

In this section we first present the improvement induced by the proposed COCOA-IH with respect to the input methods described in Section IV-C, and we then compare our results with the application of other combinational methods in the state of the art. The combinational methods belong to the three categories identified in Section II: direct combination using unsupervised combination (DC-UC), direct combination using supervised combination (DC-SC), and guided combination (GC). In order to perform a fair comparison, all the compared methods consider the same set of input methods (and parameters) as our COCOA-IH solution. The only exception is the Multi-Cue (MC) method by Li et al. [17], whose code is not available for reproduction, and whose reported results are based on the same methods although with slight variation in the choice of parameters.

TABLE II
RESULTS OF COMBINATIONAL SINGLE-IMAGE ILLUMINANT ESTIMATION ALGORITHMS, IN TERMS OF ANGULAR ERROR (DEGREES) ON THE SHI-GEHLER DATASET, AND COMPARISON WITH THE COMBINATIONAL ALGORITHMS IN THE STATE OF THE ART. ALGORITHMS ARE DIVIDED INTO DIRECT COMBINATION WITH UNSUPERVISED COMBINATION (DC-UC), DIRECT COMBINATION WITH SUPERVISED COMBINATION (DC-SC), AND GUIDED COMBINATION (GC)

	Method	Mean	Med.	Trim.	95 Pctl.	Max
Input	Shades of Gray (SoG) (0,4,0)	4.58	2.58	3.26	14.18	22.79
	Gray World (GW) (0,1,0)	4.78	3.65	3.94	10.76	24.91
	Gray Edge 1st order (GE1) (1,1,6)	3.94	2.85	3.14	11.70	23.37
	Gray Edge 1st order (GE1-fast) (1,1,1)	4.09	3.15	3.40	10.44	18.91
	Gray Edge 2nd order (GE2) (2,1,1)	4.12	3.31	3.51	10.41	17.77
	general Gray World (gGW) (0,9,9)	4.40	2.89	3.30	14.07	22.40
	general Gray World (gGW-fast) (0,1,1)	4.79	3.67	3.96	13.42	25.03
	White Patch (WP) (0,∞,0)	6.36	3.93	4.68	19.37	45.78
DC-UC	Simple Committee [5]	4.18	3.00	3.42	11.15	20.55
	Nearest-2 (global) (N2) [6]	3.93	2.88	3.15	11.00	19.99
	Nearest-2 (per image) (N2) [6]	4.04	2.54	2.95	13.27	22.07
	Nearest-10% (global) (N-10%) [6]	3.98	2.63	3.07	11.73	20.80
	Nearest-10% (per image) (N-10%) [6]	4.01	2.55	2.96	12.84	21.97
	Nearest-30% (global) (N-30%) [6]	3.98	2.68	3.07	11.90	21.49
	Nearest-30% (per image) (N-30%) [6]	4.02	2.65	3.01	12.84	22.65
	No-1-max (global) (N1M) [6]	4.03	2.84	3.27	11.49	20.57
	No-1-max (per image) (N1M) [6]	3.96	2.71	3.10	12.15	21.32
	No-2-max (global) (N2M) [6]	3.98	2.63	3.07	11.73	20.80
	No-2-max (per image) (N2M) [6]	3.90	2.49	2.98	11.63	20.83
	Median (global) (MD) [6]	3.94	2.85	3.14	11.70	23.37
	Median (per image) (MD) [6]	3.89	2.66	3.03	11.71	20.83
DC-SC	LMS Committee [5]	4.27	2.62	2.95	11.95	68.72
	Extreme Learning Machine (ELM) [7]	4.10	3.01	3.17	12.15	22.71
	Support Vector Regr. (lin) (SVRL) [7]	3.51	2.87	2.96	9.17	16.51
	Support Vector Regr. (rbf) (SVRR) [7]	3.26	2.45	2.61	9.32	18.16
	COCOA-IH (this work)	2.66	1.78	1.95	8.54	21.45
	COCOA-IH-fast (this work)	2.70	1.77	1.96	8.84	17.71
GC	Natural Image Statistics comb. (NIS) [14]	4.07	2.98			20.37
	Bianco et al. 2010 [13]	4.09	2.93			20.44
	Bianco et al. 2008 [12]	3.89	2.63			20.68
	Multi-Cue (MC) [17]	3.25	2.20	2.55		

The results of illuminant estimation on the Shi-Gehler dataset are reported in terms of average, median, trimean, 95th percentile and maximum angular error statistics in Table II, comparing with other combinational illuminant estimation methods. Additional comparisons with state-of-the-art illuminant estimation methods are reported in Table III and commented in the following Section. Our COCOA-IH model is able to reduce by 32% the mean angular error with respect to the best input method (GE1) and by 58% with respect to the worst one (WP), thus suggesting a good ability at feature selection and combination. An in-depth analysis of the impact of each underlying input method is provided in Section V-E. From the reported results it is possible to see that COCOA-IH is also able to outperform by a large margin the other compared combinational methods belonging to all analyzed groups. The version of our model with fast parameters, COCOA-IH-fast, produces generally equivalent results with respect to COCOA-IH in this setup.

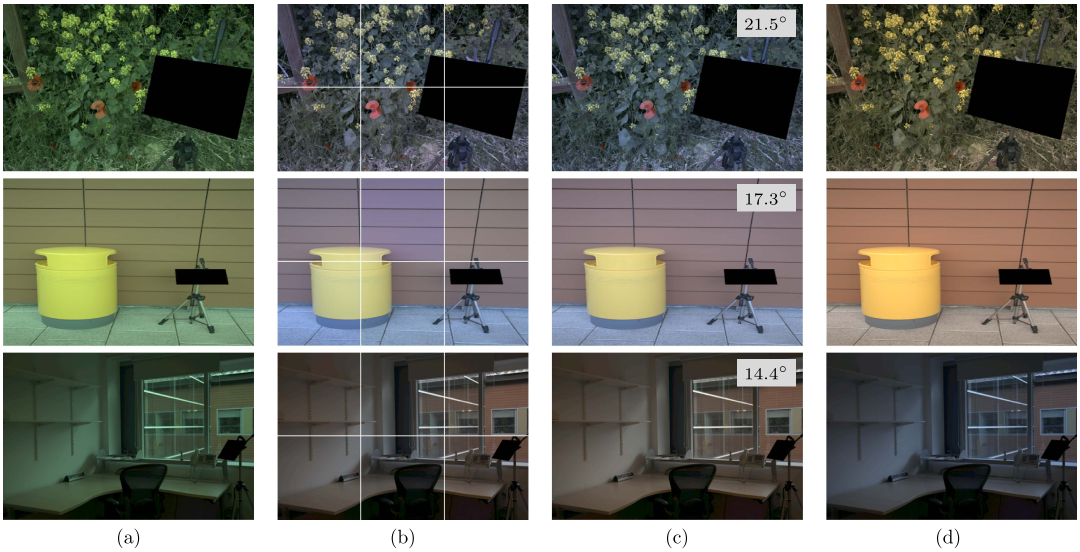


Fig. 3. Visualization of the three images of the Shi-Gehler dataset on which COCOA-IH obtains the three worst results. Input image (a); collage image obtained from the six images respectively collected the illuminant estimated by each of the six individual algorithms (b); image corrected with the illuminant estimated by COCOA-IH, with the angular error overlaid in the top right corner (c); ground truth, i.e. image corrected with the ground truth illuminant (d).

TABLE III
Comparison in Terms of Angular Error (Degrees) With the Individual, Single-Image Illuminant Estimation Algorithms in the State of the Art on the Shi-Gehler and NUS Datasets. As a Subscript to All the Mean and Median Angular Values, It is Reported Its Position in a Hypothetical Ranking

		ColorChecker			NUS		
		Mean	Median	Max	Mean	Median	Max
Param.	Bright Pixels (BP) [29]	$3.98_{(21)}$	$2.61_{(19)}$				
	Cheng et al. [30]	$3.52_{(19)}$	$2.14_{(16)}$	28.35	$3.02_{(11)}$	$2.12_{(11)}$	23.28
	Grey Pixel (edge) [31]	$4.60_{(23)}$	$3.10_{(22)}$		$3.15_{(12)}$	$2.20_{(13)}$	
	Grey Pixel (revised) [47]	$3.07_{(16)}$	$1.87_{(12)}$				
Unsup.	Buzzelli et al. (gl. norm) [32]	$4.84_{(25)}$	$4.12_{(25)}$	20.80	$4.88_{(14)}$	$4.17_{(20)}$	18.7
	Buzzelli et al. (ch. norm) [32]	$5.48_{(26)}$	$4.81_{(26)}$	19.88	$4.32_{(13)}$	$3.37_{(19)}$	22.36
	Quasi-Unsupervised [33]	$3.46_{(17)}$	$2.23_{(17)}$	21.17	$3.00_{(10)}$	$2.25_{(14)}$	19.16
Supervised	Bayesian [25]	$4.70_{(24)}$	$3.44_{(24)}$			$2.81_{(18)}$	
	Spatio-Spectral (ML) [34]	$3.55_{(20)}$	$2.93_{(21)}$			$2.54_{(16)}$	
	Spatio-Spectral (GP) [34]	$3.47_{(18)}$	$2.90_{(20)}$			$2.39_{(15)}$	
	Natural Image Statistics [14]	$4.09_{(22)}$	$3.13_{(23)}$			$2.69_{(17)}$	
	Exemplar-based [35]	$2.89_{(13)}$	$2.27_{(18)}$				
	Chakrabarti (Empirical) [36]	$2.89_{(13)}$	$1.89_{(13)}$				
	Chakrabarti (End-to-end) [36]	$2.56_{(9)}$	$1.67_{(9)}$				
	Cheng et al. [37]	$2.42_{(8)}$	$1.65_{(8)}$			$1.58_{(7)}$	
	Bianco et al. [38]	$2.36_{(7)}$	$1.44_{(6)}$	16.98		$1.77_{(8)}$	
	FFCC [8]	$1.78_{(3)}$	$0.96_{(1)}$	16.25	$1.99_{(2)}$	$1.34_{(1)}$	19.8
	Oh and Kim [39]	$2.16_{(6)}$	$1.47_{(7)}$		$2.41_{(8)}$	$2.15_{(12)}$	
	CCC (dist+ext) [40]	$1.95_{(5)}$	$1.22_{(5)}$		$2.38_{(6)}$	$1.48_{(4)}$	
	FC4 (AlexNet) [27]	$1.77_{(2)}$	$1.11_{(3)}$		$2.12_{(4)}$	$1.53_{(6)}$	
	DS-Net (HypNet+SelNet) [41]	$1.90_{(4)}$	$1.12_{(4)}$		$2.24_{(5)}$	$1.46_{(3)}$	
	Quasi-Unsupervised + Fine Tune [33]	$2.91_{(15)}$	$1.98_{(15)}$	19.90	$1.97_{(1)}$	$1.41_{(2)}$	20.5
	SIIE [48]	$2.77_{(12)}$	$1.93_{(14)}$	18.45	$2.05_{(3)}$	$1.50_{(5)}$	
	COCOA-IH (this work)	$2.66_{(10)}$	$1.78_{(11)}$	21.45	$2.41_{(9)}$	$1.83_{(10)}$	18.47
	COCOA-IH-fast (this work)	$2.70_{(11)}$	$1.77_{(10)}$	17.71	$2.38_{(10)}$	$1.79_{(9)}$	16.10
	COCOA-IH-advanced (this work)	$1.60_{(1)}$	$1.04_{(2)}$	14.32			

In Fig. 3 we report the three images of the Shi-Gehler dataset on which COCOA-IH obtains the worst results, while the three images on which it obtains the best results are reported in Fig. 4. It is possible to notice how the worst results correspond to images with colored background/objects and to a scene with multiple illuminants. The best results instead correspond to images in which the underlying assumptions of the individual methods used by COCOA-IH are more likely to be satisfied.

To further analyze the performance of the proposed combination framework we trained COCOA-IH with reduced versions of the training set of the Shi-Gehler dataset. The sizes considered are determined by successively halving its original size from 1, corresponding to the original size, to $1/32$. For each training set size, five different random selections (runs) are performed: in Fig. 5 we plot the average angular error and its standard deviation for each trained model, averaged over the different runs performed. In the same plot we report the performance of the best input algorithm combined by COCOA-IH (i.e. Gray Edge 1st order) as a dashed line. As can be seen in the plot the best performance are obtained when all the data available for training in the original splits of the Shi-Gehler dataset are used (i.e. about 378 images, averaged over the three cross validation folds). As expected as we reduce the training set size the average angular error increases. Nevertheless, even reducing the training set to $1/8$ of its original size, which corresponds to a total of about 48 images (to be further split into the actual training set and validation set according to a 80%-20% ratio), COCOA-IH still performs better than the best input method combined. For smaller training sets the average angular error rapidly degrades, showing no advantage of using COCOA-IH over the best input method combined for a training set size equal to $1/16$ of its original size, corresponding to a total of about 24 images to be further divided into train and validation. The performed experiment shows how the proposed method COCOA-IH can improve over the best input method, even when the number of images available for training is scarce.

B. State-of-the-Art Single-Image Illuminant Estimation

In this experiment we compare the performance of the proposed COCOA-IH with respect to individual state of the art algorithms for single-image illuminant estimation on the Shi-Gehler dataset. The 21 compared methods belong to three different groups on the basis of the type and level of training they need. The first group encompasses the parametric

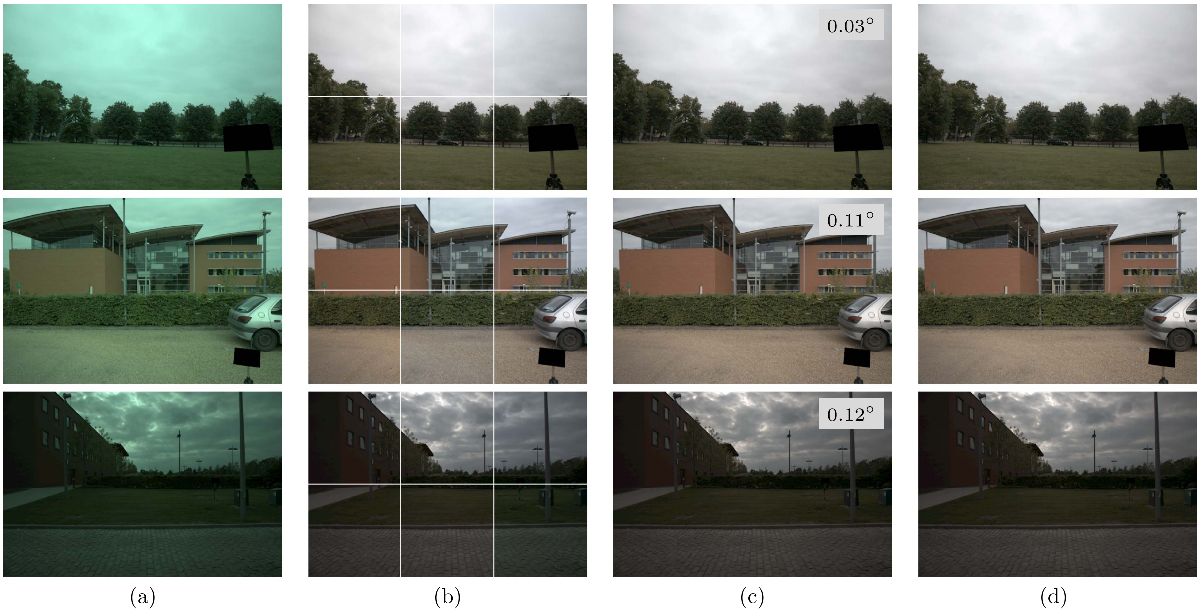


Fig. 4. Visualization of the three images of the Shi-Gehler dataset on which COCOA-IH obtains the three best results. Input image (a); collage image obtained from the six images respectively collected the illuminant estimated by each of the six individual algorithms (b); image corrected with the illuminant estimated by COCOA-IH, with the angular error overlaid in the top right corner (c); ground truth, i.e. image corrected with the ground truth illuminant (d).

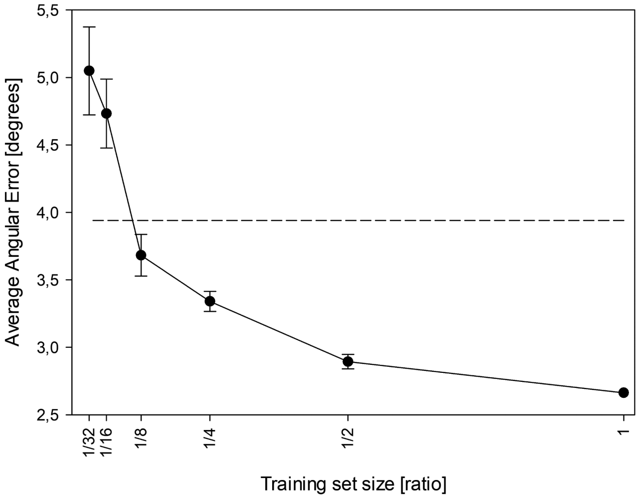


Fig. 5. Performance of COCOA-IH in terms of average angular error reducing the training set size as a ratio of the classical data partition of the Shi-Gehler dataset. The dashed line represents the performance of the best input algorithm used by COCOA-IH, i.e. Gray Edge 1st order (GE1).

methods: Bright Pixels (BP) [29], Cheng et al. [30], and Grey Pixel (edge) [31]. In the second group there are learning-based methods that require no supervision in terms of illuminant ground truth: Buzzelli et al. (global normalization and channel normalization) [32], and Quasi-Unsupervised [33]. The third group comprises the fully-supervised methods, that need a complete training on illuminant data to properly operate: Bayesian [25], Spatio-Spectral (ML and GP) [34], Natural Image Statistics [14], Exemplar-based [35], Chakrabarti (Empirical and End-to-end) [36], Cheng et al. [37], Bianco et al. [38], FFCC [8], Oh and Kim [39], CCC (dist+ext) [40], FC4 (AlexNet) [27], DS-Net (HypNet+SelNet) [41], and Quasi-Unsupervised with Fine Tuning [33].

The results in terms of average, median and maximum angular error statistics are reported in Table III. It is possible to notice how the best results are obtained within the group of supervised algorithms. The proposed method with Image Head, i.e. COCOA-IH, compares favorably with the state of the art, placing itself in the upper part of an hypothetical ranking, close to some early CNN-based methods, despite it only combines unsupervised and parametric methods.

In addition to comparing methods across multiple statistics (mean, median, maximum errors), an ideal assessment would involve the Wilcoxon signed-rank test [42] to compare the entire error distributions, and thus provide a level of statistical significance. This was, however, not possible due to the unavailability of illuminant estimations for the compared methods (aggregate statistics are reported from the corresponding publications). On the other hand, it is possible to observe that, according to a literature survey by Gijsenij et al. [43], a deviation of 1° in angular error with the ground truth is considered below the level of what can be perceived by a human being [44], while the range between 2° and 3° is considered detectable but still acceptable [45], [46].

Our current experimental setup has been specifically designed to exploit lightweight methods, resulting in a good compromise between efficiency and effectiveness. However, the same approach can be effectively applied to combine more accurate (but also more computationally-demanding) methods, if the underlying application values accuracy over speed. To this extent, we provide an example of the combination of more advanced color constancy methods from Table III for which either the code or the estimations on the Shi-Gehler dataset are publicly available: FC4 [27], FFCC [8], DS-Net [41], the method by Chakrabarti et al. (end-to-end) [36], SIIE [48], and QU (fine tune) [33]. The results show that, by focusing on the combination of more advanced methods, it is also possible to exploit their specific features to achieve state of the art results in illuminant estimation. This configuration, named COCOA-IH-advanced, could potentially be applied to other datasets, such as the NUS for images and BCC for videos, as long as estimations from the underlying methods are available.

C. *Exploiting the Temporal Component*

We investigate several solutions to exploit the temporal component, and thus to process video sequences. In general, they can be classified as embedding the temporal component before or after the combination of input methods. Combining before (B) allows exploiting the temporal component in each single input method, while combining after (A) means exploiting the temporal information only once, at the combination level.

The investigated solutions are the following.

- Frame average: it is the simplest approach where the output illuminant for a video corresponds to the average of the estimates on each frame. If it is applied before (B) the combination, the estimates of each single input algorithm are individually averaged to give the corresponding video illuminant estimate; these estimates are then combined by COCOA-IH to give the final estimate. If it is applied after (A) the combination, COCOA-IH is applied to the estimates given by the individual methods to each frame, and the estimates by COCOA-IH for each frame are then averaged to give the final estimate.
- Frame median: it is the same approach as the previous one but considering the median instead of the average operations to combine the per-frame estimates.
- Gaussian weights with free standard deviation: it is an extension of the first approach, where the combination weights are not uniform anymore but are taken from a Gaussian distribution with a free standard deviation σ. The Gaussian distribution is centered on the last frame and therefore decreasing weights are given to the older frames. For simplicity, all the sequences are extended to a common length by adding the necessary number of dummy illuminant estimates at the beginning of each sequence.
- Gaussian weights with free standard deviation and center: it is an extension of the previous approach, in which we let also the center x_0 of the Gaussian to be a free parameter. Similarly to the previous approach, all the sequences are extended to a common length.
- LSTM - Long Short-Term Memory: in this approach the temporal component is exploited using LSTMs. When LSTMs are used before (B) the combination, one LSTM is applied to each of the inputs of COCOA-IH and the resulting model is trained end to end. When the temporal component is exploited after (A) the combination, a single LSTM is used and the model corresponds to the COCOA-VH described in Section III. LSTM (B) has been initialized with the same configuration as LSTM (A). The main architectural difference is in the input and output feature size that in LSTM (B) have been set to 3, corresponding to the dimensionality of the input estimations to be time processed.

TABLE IV
COMPARISON OF DIFFERENT SOLUTIONS TO EXPLOIT THE TEMPORAL COMPONENT, TESTED ON THE BCC DATASET. THE "TIME" COLUMN REFERS TO BEFORE (B) OR AFTER (A) THE COMBINATION OF INPUT METHODS

Method	Time	Mean	Med.	95 Pctl.
Frame average	B	4.20	3.15	12.32
Frame median	B	4.10	2.68	13.12
Gauss. weights (σ)	B	4.23	3.03	12.37
Gauss. weights (σ, x_0)	B	3.98	2.90	11.51
LSTM	B	2.77	2.06	8.46
Frame average	A	2.83	2.11	7.79
Frame median	A	2.88	2.05	9.12
Gauss. weights (σ)	A	2.88	2.17	7.82
Gauss. weights (σ, x_0)	A	2.67	1.91	8.08
LSTM (**COCOA-VH**)	A	2.61	1.66	8.81

TABLE V
COMPARISON IN TERMS OF ANGULAR ERROR WITH THE VIDEO ILLUMINANT ESTIMATION ALGORITHMS IN THE STATE OF THE ART ON THE BCC DATASET

Method	Mean	Med.	95 Pctl.
Prinet et al. [21]	7.51	6.94	20.70
Temporal extended GI (T.GI from [10])	4.73	2.96	17.42
Temporal extended FFCC [8]	3.35	1.70	17.41
RCC-Net [23]	2.74	2.23	8.21
BCC-Net [10]	1.99	1.21	6.34
COCOA-VH (this work)	2.61	1.66	8.81
COCOA-VH-fast (this work)	2.66	1.88	8.44

The different approaches considered to exploit the temporal component are tested on the BCC dataset [10]. The numerical results of this comparison are reported in Table IV: it is possible to see that the best performance in terms of both average and median angular errors are obtained by the COCOA-VH which uses an LSTM to exploit the temporal component. More in general, it is possible to see how the approaches that exploit the temporal component after the combination (A) obtain better results than the corresponding versions that exploit it before (B) the combination: on average this improvement is 1.1 degrees on the mean angular error, 0.8 degrees on the median angular error and 3.2 on the maximum angular error, respectively corresponding to a 26.6%, 27.7% and 25.9% improvement.

D. *State-of-the-Art Video Illuminant Estimation*

In this experiment we compare the proposed COCOA-VH against state-of-the art video illuminant estimation methods on the BCC dataset. We focus on methods specifically designed for videos/image sequences (i.e. Prinet et al. [21], RCC-Net [23] and BCC-Net [10]), as well as two existing temporal extensions of supervised single-frame algorithms (i.e. T.GI for Grayness Index [47] and T.FFCC for Fast Fourier Color Constancy [8]).

The numerical results in terms of average, median and 95th percentile angular error statistics are reported in Table V. The results show how the proposed COCOA-VH ranks second in

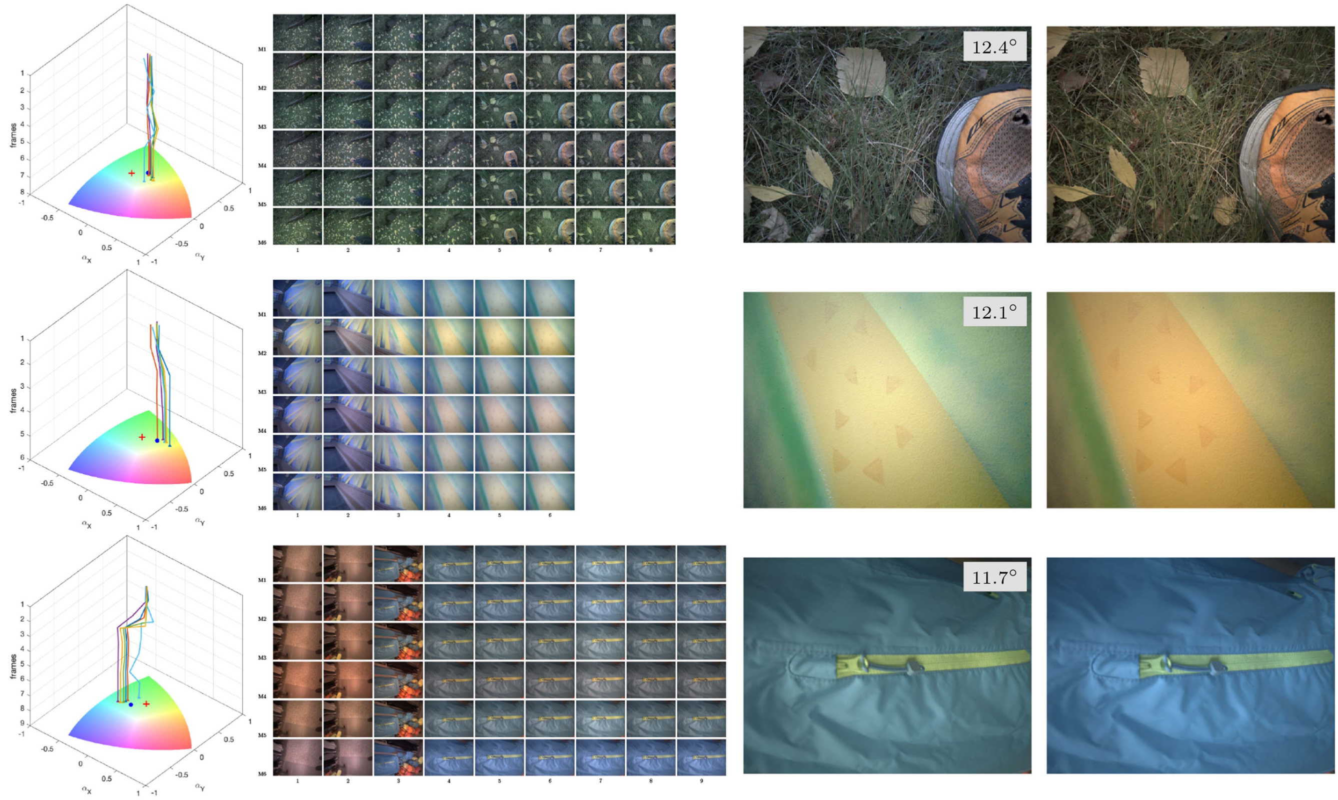


Fig. 6. Worst 3 results of COCOA-VH on the BCC dataset. Column 1: plot of the estimates given by the different combined algorithms across the sequence; the blue dot represents the illuminant estimated on the shot frame by COCOA-VH, while the red cross represents the ground truth. Column 2: sequence frames corrected with the estimate given by the different combined algorithms, respectively SoG, GE1, GE2, GGW, GW, and WP. Column 3: shot frame corrected with the estimate by COCOA-VH. Column 4: shot frame corrected with the ground truth illuminant.

terms of both the average and the median error, surpassing more complex methods.

In Fig. 6 we report the three sequences of the BCC dataset on which COCOA-VH obtains the three worst results. For each sequence we draw the plot of the illuminant estimated by each of the six combined algorithms on each frame of the sequence plotted as chromaticities in the ARC space [49] together with the final estimate by COCOA-VH and the ground truth. The plots show how in the initial frames the six estimates are closer to the ground truth and then start to diverge from it, thus causing the drift of the final COCOA-VH estimate. In Fig. 7 we report the three sequences of the BCC dataset on which COCOA-VH obtains the three best results. From the plots it is possible to notice how these cases correspond to sequences on which the combined algorithms already provide a good illuminant estimate. Concerning the content of the sequences obtaining the worst and best results we can observe a strong similarity with those reported in Figs. 3 and 4. This is not surprising since both COCOA-IH and COCOA-VH exploit the same set of input illuminant estimation methods and they have the same backbone architecture, just differing in the regression head.

As a further analysis we measure the computational complexity of the compared methods, focusing on video illuminant estimation due to the critical role that efficiency assumes in this domain: fast online processing allows a direct feedback in the camera viewfinder, and fast offline processing enables handling large amounts of video data. Given the heterogeneous nature of the code available for the different methods, and the different hardware on which they run (i.e. CPU vs GPU), in order to perform a fair comparison we decided to calculate the number of floating point operations for each compared method. In Fig. 8 we plot the average angular error reached by each method reported in Table V with respect to the number of operations. From the plot we can observe how the proposed methods are in the bottom left corner of the plot, providing the best trade-off between illuminant estimation accuracy and computational complexity, with COCOA-VH-fast being the one requiring the lowest number of operations, i.e. 16.6 millions of operations (M-Ops) of which just 0.56% are due to the actual nonlinear combination. In practice, COCOA-VH and COCOA-VH-fast work at 21.97 FPS and 31.48 FPS respectively, the latter fully reaching the real-time threshold of 30 FPS [8], [9], with the bottleneck being the CPU-based implementation of the input methods. A lower illuminant estimation error is obtained by BCC-Net [10], that requires a number of operations that is two orders of magnitude higher, i.e. 3277.1 M-Ops.

E. Sensitivity Analysis

In this experiment we carry out a sensitivity analysis of COCOA-IH in order to understand how a change in one of the six inputs affects the output.

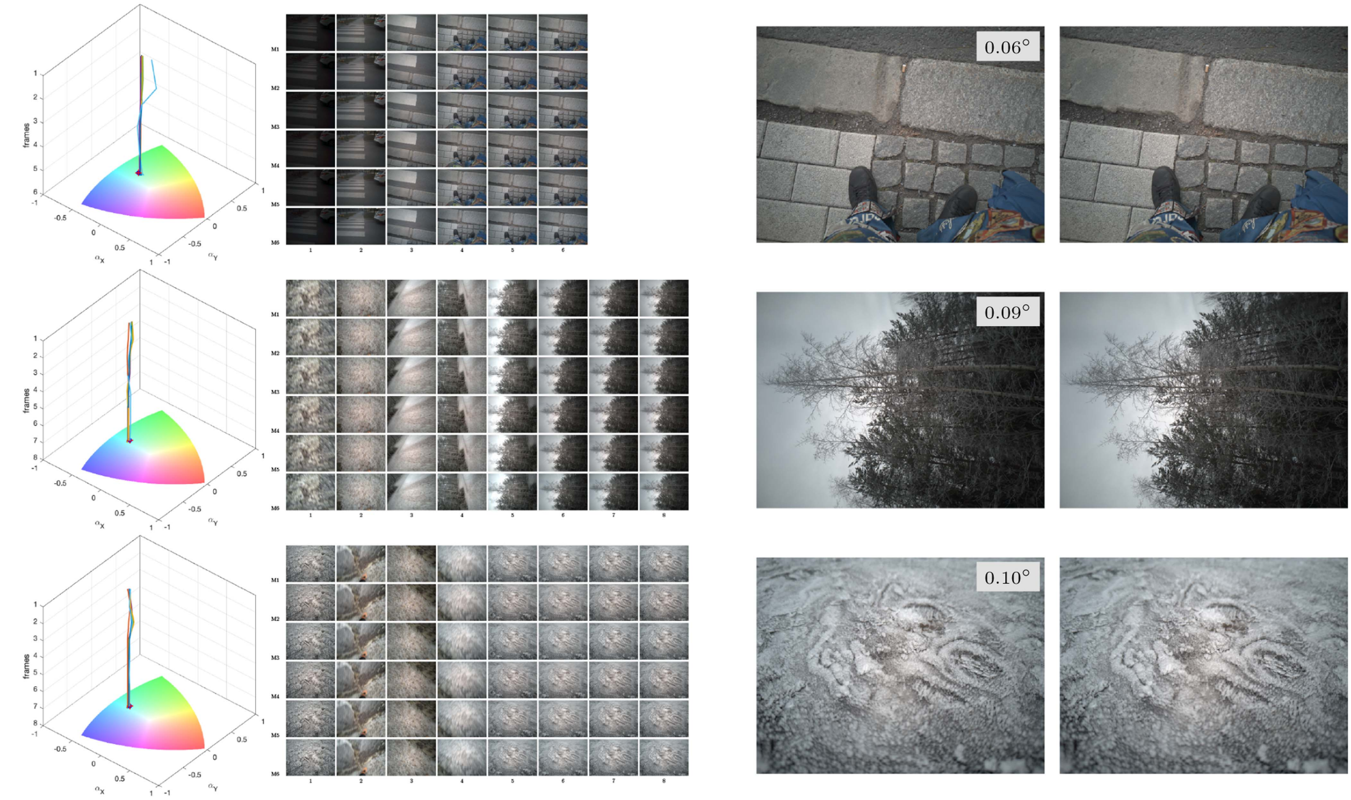


Fig. 7. Best 3 results of COCOA-VH on the BCC dataset. Column 1: plot of the estimates given by the different combined algorithms across the sequence; the blue dot represents the illuminant estimated on the shot frame by COCOA-VH, while the red cross represents the ground truth. Column 2: sequence frames corrected with the estimate given by the different combined algorithms, respectively SoG, GE1, GE2, GGW, GW, and WP. Column 3: shot frame corrected with the estimate by COCOA-VH. Column 4: shot frame corrected with the ground truth illuminant.

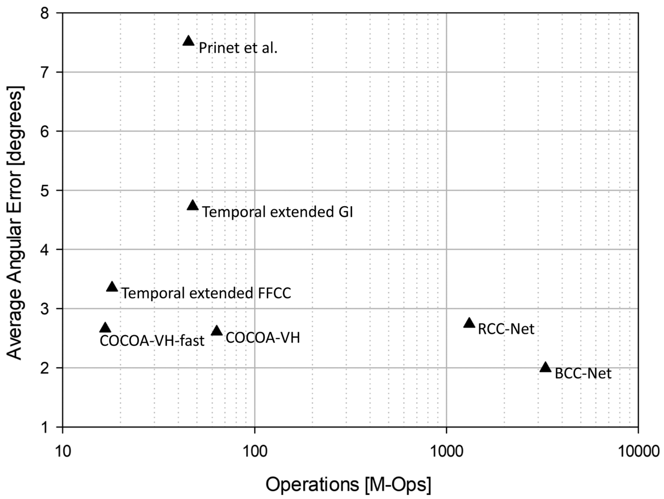


Fig. 8. Plot representing the average angular error (in degrees) with respect to the computational complexity (in terms of millions of operations) of the methods reported in Table V. The ideal point is in the bottom-left corner.

The sensitivity analysis is performed exploiting the ARC color space [49], that has the property that euclidean distances correspond to angular errors. We modified each of the six inputs individually, by modifying the estimate given by the corresponding algorithm from 0 to 0.1 radians (i.e. approximately 5.7°) in steps of 0.01 radians. The modification is performed along 36 directions, in order to cover the possible hues in 10° steps. The considered datasets are the reprocessed Shi-Gehler [25] and the NUS [30], and for each possible input modification the average angular error is computed. The six surfaces obtained by considering all the possible input modifications of each input individually are reported in the top row of Fig. 9 for Shi-Gehler, and the top row of Fig. 10 for NUS. The bottom row of the same figures report the level curves of these surfaces. In all the plots the corresponding crop of the ARC space is reported as a reference in order to understand the sensitivity with respect to different hues. The center point of all the six plots correspond to the case where no input is modified and thus the result corresponds to the average angular error reported in Tables III and II for COCOA-IH (i.e. 2.66°).

From the reported plots it is possible to notice how in general there are inputs with respect to which COCOA-IH is more sensitive, i.e. the third and the fifth inputs respectively corresponding to GE2 and GW. This is also numerically confirmed in Table VI where the average slope for each surface is computed. Furthermore we can observe how the sensitivity is not isotropic for any of the inputs, but the surfaces are approximately symmetric with an axis of symmetry passing close to the center of the plot and with a different direction for each of the inputs. The approximate direction of the axis of symmetry is reported for each surface in Table VI. In particular COCOA-IH is very

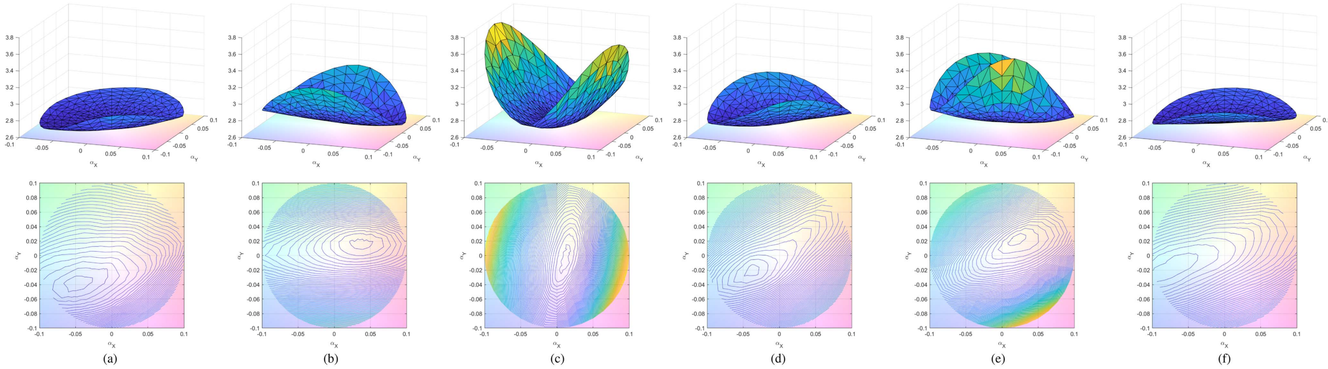


Fig. 9. Sensitivity analysis of COCOA-IH with respect to the six inputs individually. From left to right: SoG (a), GE1 (b), GE2 (c), GGW (d), GW (e), and WP (f). Top row: surface representing how the average angular error on Shi-Gehler dataset changes when the corresponding input is modified. Bottom row: level curves of the corresponding surfaces in the top row.

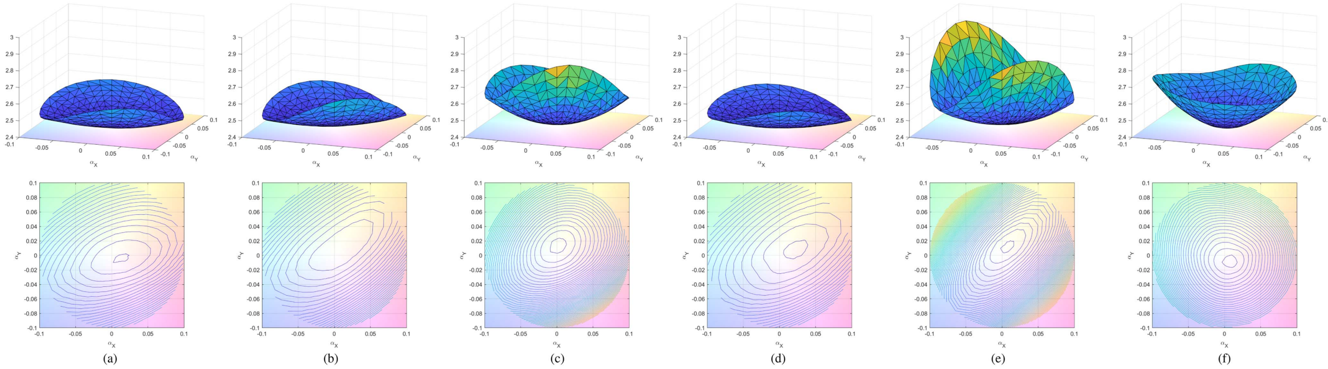


Fig. 10. Sensitivity analysis on NUS dataset of COCOA-IH with respect to the six inputs individually. From left to right: SoG (a), GE1 (b), GE2 (c), GGW (d), GW (e), and WP (f). Top row: surface representing how the average angular error on NUS dataset changes when the corresponding input is modified. Bottom row: level curves of the corresponding surfaces in the top row.

TABLE VI

Statistics of the Sensitivity Analysis of the COCOA-IH Model With Respect to the Individual Inputs on Shi-Gehler Dataset (Left) and on NUS Dataset (Right): Average Slope, the Higher the More Sensitive is the Model With Respect to the Corresponding Input. Direction of Axis of Symmetry, That Approximately Corresponds to the Direction of Lowest Sensitivity

Input	Avg. slope	Axis of symm.	Avg. slope	Axis of symm.
SoG	1.4410	30°	1.2218	20°
GE1	2.9751	10°	1.2795	40°
GE2	6.1226	80°	2.5606	50°
gGW	2.3920	30°	1.0348	30°
GW	4.3267	30°	3.1130	50°
WP	1.4736	20°	2.4321	130°

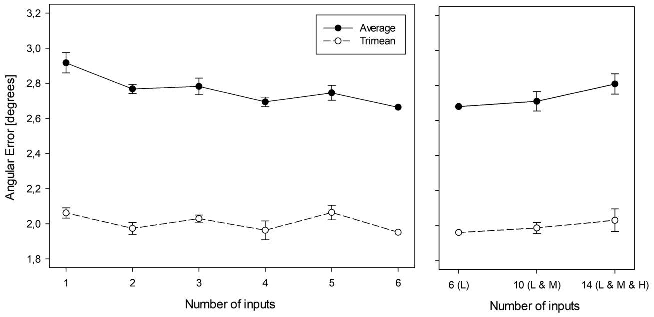
sensitive to changes in the red-cyan direction for what concerns GE2 with an axis of symmetry approximately oriented at 80°, while the most sensitive direction with respect to GW is the green-purple direction with an axis of symmetry approximately oriented at 30°. We can also observe how COCOA-IH has a very low sensitivity with respect to the first and the sixth inputs, i.e. SoG and WP. It is also possible to notice how there is a region for each input able to obtain a lower average angular error with respect to the one obtained when no change is applied to the inputs. This is due to the fact that in the classical three-fold subdivision of the Shi-Gehler dataset, the training and testing illuminants have a different distribution.

F. *Performance Analysis With Different Input Cardinalities*

In this experiment we want to investigate the behavior of COCOA-IH when a different number of inputs is available.

We start considering the case when a lower number of inputs is available: starting from COCOA-IH we remove one input at a time in order of increasing sensitivity (i.e. the one with the lowest sensitivity is removed first). The performance in terms of average and median angular errors are reported in Fig. 11 a. It is possible to observe a general trend of the average error increasing with a lower number of inputs, while the trimean error shows an oscillatory behavior.

To consider the case of more input methods, we identified three categories for the values that the free parameters p and σ in Table I can assume: 1 (Low, exactly corresponding to the COCOA-IH-fast configuration), 5 (Medium), and 9 (High). Fig. 11 b thus shows the impact on the average and trimean recovery error statistics by considering an ever increasing



(a)

(b)

Fig. 11. Performance analysis of COCOA-IH by reducing (a), and increasing (b) the number of inputs.

number of input methods: Low (L); Low and Medium (L&M); Low, Medium, and High (L&M&H). It is possible to observe a general trend of the error increasing with the number of inputs. These results suggest that additional variations of the same input methods do not provide any added value, as the corresponding estimations are possibly highly correlated. We therefore hypothesize as future development the formulation of input feature selection. Furthermore, an optimization of the network architecture based on different input cardinalities could allow for a better exploitation of such inputs.

VI. Conclusion

Computational color constancy has been addressed through the years with a wide variety of approaches, often relying on different assumptions over the input image. These approaches are increasingly computation demanding, memory demanding, and data greedy. In this paper we have proposed a fusion strategy that efficiently exploits a variety of simple learning-free algorithms for computational color constancy, combining them in order to provide a lightweight solution that still achieves high performance. Our solution, which can be specialized to either the image domain or the video domain, has been thoroughly evaluated in a wide range of experimental setups on standard benchmark datasets. We have compared our combination strategy for still images against other combining solutions achieving top performance, and reaching an illuminant estimation accuracy comparable to more sophisticated solutions. We have also explored different solutions to exploit the temporal component available when analyzing a full video sequence, and experimentally defined a version of our model that exploits a LSTM module to handle varying-length videos. This solution has been tested against other algorithms for video color constancy, both in terms of angular error and computational complexity, achieving state-of-the-art performance. Knowing that adaptation to new devices is a real need in the application domain, we have shown that reducing the number of training images with respect to the standard dataset partition, our method is still able to effectively combine the input methods. Finally, we have conducted a sensitivity analysis aimed at interpreting the combination strategy learned by our model, and understanding how a change in the inputs affect the output.

As future developments, we intend to further explore the possibilities of input combination when dealing with different camera sensors, as well as the combination of more complex input algorithms to further reduce the illuminant estimation error. Furthermore, in this paper we focused on color constancy methods that are based on the assumption of a single illuminant, i.e. producing a single estimate per image. However, a combination technique similar to the proposed one could be applied to methods designed for multiple illuminant estimation, possibly under the constraint that they produce estimations in a consistent form.

Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

References

[1] 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.

[2] M. A. Hussain, A. Sheikh-Akbari, and E. A. Halpin, “Color constancy for uniform and non-uniform illuminant using image texture,” *IEEE Access*, vol. 7, pp. 72964–72978, 2019.

[3] S. Bianco, G. Ciocca, and R. Schettini, “Combination of video change detection algorithms by genetic programming,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 914–928, Dec. 2017.

[4] S. Bianco, M. Buzzelli, G. Ciocca, and R. Schettini, “Neural architecture search for image saliency fusion,” *Inf. Fusion*, vol. 57, pp. 89–101, 2020.

[5] V. C. Cardei and B. Funt, “Committee-based color constancy,” in *Proc. Color Imag. Conf.*, 1999, pp. 311–313.

[6] S. Bianco, F. Gasparini, and R. Schettini, “Consensus-based framework for illuminant chromaticity estimation,” *J. Electron. Imag.*, vol. 17, no. 2, 2008, Art. no. 23013.

[7] B. Li, W. Xiong, D. Xu, and H. Bao, “A supervised combination strategy for illumination chromaticity estimation,” *ACM Trans. Appl. Percep.*, vol. 8, no. 1, pp. 1–17, 2010.

[8] J. T. Barron and Y.-T. Tsai, “Fast fourier color constancy,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 886–894.

[9] D. Mazzini, M. Buzzelli, D. P. Pauy, and R. Schettini, “A CNN architecture for efficient semantic segmentation of street scenes,” in *Proc. IEEE 8th Int. Conf. Consum. Electron.*, 2018, pp. 1–6.

[10] Y. Qian, J. Käpylä, J.-K. Kämäräinen, S. Koskinen, and J. Matas, “A benchmark for burst color constancy,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 359–375.

[11] B. Li, W. Xiong, W. Hu, and B. Funt, “Evaluating combinational illumination estimation methods on real-world images,” *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1194–1209, Mar. 2014.

[12] 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.

[13] 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, 2010.

[14] A. Gijsenij and T. Gevers, “Color constancy using natural image statistics and scene semantics,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 687–698, Apr. 2011.

[15] J. Van De Weijer, C. Schmid, and J. Verbeek, “Using high-level visual information for color constancy,” in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

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

[17] B. Li, W. Xiong, W. Hu, B. Funt, and J. Xing, “Multi-cue illumination estimation via a tree-structured group joint sparse representation,” *Int. J. Comput. Vis.*, vol. 117, no. 1, pp. 21–47, 2016.

[18] S. K. Subhashdas, Y.-H. Ha, and D.-H. Choi, “Multi-class dynamic weight model for combinational color constancy,” *J. Imag. Sci. Technol.*, vol. 62, no. 3, pp. 030502-1–030502-17, 2018.

[19] S. K. Subhashdas, Y.-H. Ha, and D.-H. Choi, “Hybrid direct combination color constancy algorithm using ensemble of classifier,” *Expert Syst. Appl.*, vol. 116, pp. 410–429, 2019.

[20] Q. Yang, S. Wang, N. Ahuja, and R. Yang, "A uniform framework for estimating illumination chromaticity, correspondence, and specular reflection," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 53–63, Jan. 2010.

[21] V. Prinet, D. Lischinski, and M. Werman, "Illuminant chromaticity from image sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3320–3327.

[22] N. Wang, B. Funt, C. Lang, and D. Xu, "Video-based illumination estimation," in *Proc. Int. Workshop Comput. Color Imag.*, 2011, pp. 188–198.

[23] Y. Qian, K. Chen, J. Nikkanen, J.-K. Kamarainen, and J. Matas, "Recurrent color constancy," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5458–5466.

[24] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083951076&partnerID=40&md5=1512fe7d6538ffc6686cf01cba3c3460

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

[26] L. Shi and B. Funt, "Re-processed version of the gehler color constancy dataset of 568 images," 2012, Accessed: Jul. 30, 2021. [Online]. Available: https://www2.cs.sfu.ca/~colour/data/shi_gehler/

[27] Y. Hu, B. Wang, and S. Lin, "Fc4: Fully convolutional color constancy with confidence-weighted pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4085–4094.

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

[29] 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.

[30] 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," *JOSA A*, vol. 31, no. 5, pp. 1049–1058, 2014.

[31] K.-F. Yang, S.-B. Gao, and Y.-J. Li, "Efficient illuminant estimation for color constancy using grey pixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2254–2263.

[32] M. Buzzelli, J. van de Weijer, and R. Schettini, "Learning illuminant estimation from object recognition," in *Proc. IEEE 25th Int. Conf. Image Process.*, 2018, pp. 3234–3238.

[33] S. Bianco and C. Cusano, "Quasi-unsupervised color constancy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12212–12221.

[34] 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.

[35] H. R. V. Joze and M. S. Drew, "Exemplar-based color constancy and multiple illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 860–873, May 2014.

[36] A. Chakrabarti, "Color constancy by learning to predict chromaticity from luminance," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 163–171.

[37] 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.*, 2015, pp. 1000–1008.

[38] S. Bianco, C. Cusano, and R. Schettini, "Single and multiple illuminant estimation using convolutional neural networks," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4347–4362, Sep. 2017.

[39] S. W. Oh and S. J. Kim, "Approaching the computational color constancy as a classification problem through deep learning," *Pattern Recognit.*, vol. 61, pp. 405–416, 2017.

[40] J. T. Barron, "Convolutional color constancy," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 379–387.

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

[42] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. Berlin, Germany: Springer, 1992, pp. 196–202.

[43] A. Gijsenij, T. Gevers, and M. P. Lucassen, "Perceptual analysis of distance measures for color constancy algorithms," *JOSA A*, vol. 26, no. 10, pp. 2243–2256, 2009.

[44] B. Funt, K. Barnard, and L. Martin, "Is machine colour constancy good enough?," in *Proc. Eur. Conf. Comput. Vis.*, 1998, pp. 445–459.

[45] G. D. Finlayson, S. D. Hordley, and P. Morovic, "Colour constancy using the chromagenic constraint," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 1079–1086.

[46] C. Fredembach and G. Finlayson, "Bright chromagenic algorithm for illuminant estimation," *J. Imag. Sci. Technol.*, vol. 52, no. 4, pp. 40906–40911, 2008.

[47] Y. Qian, J.-K. Kamarainen, J. Nikkanen, and J. Matas, "On finding gray pixels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8062–8070.

[48] M. Afifi and M. S. Brown, "Sensor-independent illumination estimation for DNN models," in *Proc. 30th Brit. Mach. Vis. Conf.*, 2020. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85087334182&partnerID=40&md5=7935a08b6b4858439ba3d7391a1a77ab

[49] M. Buzzelli, S. Bianco, and R. Schettini, "ARC: Angle-retaining chromaticity diagram for color constancy error analysis," *JOSA A*, vol. 37, no. 11, pp. 1721–1730, 2020.

Simone Zini received the B.Sc. and M.Sc. degrees in computer science from the Department of Informatics, Systems and Communication, University of Milano-Bicocca, Milan, Italy, in 2015 and 2018, respectively, and the Ph.D. degree from the University of Milano – Bicocca, in 2022. He is currently employed as a Postdoctoral Researcher. His research interests concern machine learning, image enhancement, and computational photography.

Marco Buzzelli received the bachelor's and master's degree in computer science respectively in 2012 and 2014, focusing on image processing and computer vision tasks, and the Ph.D. degree in computer science from the University of Milano, Bicocca, Italy, in 2019. He is currently employed as a Postdoctoral Researcher. His main research interests include characterization of digital imaging devices, and object recognition in complex scenes.

Simone Bianco is currently an Associate Professor of computer science with the University of Milano – Bicocca, Milan, Italy, holder of the Italian National Academic Qualification as a Full Professor of computer engineering (09/H1) and computer science (01/B1). He is on Stanford University's World Ranking Scientists List for his achievements in Artificial Intelligence and Image Processing. His teaching and research interests include computer vision, artificial intelligence, machine learning, optimization algorithms applied in multimodal, and multimedia applications. He is also a R&D Manager of the University of Milano Bicocca spin off Imaging and Vision Solutions, and Member of European Laboratory for Learning and Intelligent Systems.

Raimondo Schettini is currently a Full Professor with the University of Milano Bicocca, Milan, Italy, where he leads the Imaging and Vision Lab. Since 1987, he has been associated with the Italian National Research Council, where he led the Color Imaging Lab from 1990 to 2002. He is a team Leader in several research projects, some of them supported by companies. He authored or coauthored more than 400 refereed papers and 12 patents about color imaging, image processing, analysis, and classification, image, and video understanding and retrieval. He supervised more than ten Ph.D. students. He is the Chair of several international conferences and workshops and he is the Editor in Chief of the MDPI Journal of Imaging. He is on Stanford University's World Ranking Scientists List for his achievements in artificial intelligence and image processing. He is a Fellow of the International Association of Pattern Recognition for his contributions to pattern recognition research and color image analysis, and Fellow of Asia-Pacific Artificial Intelligence Association. Raimondo Schettini is also the Chief Technical Officer of the University of Milano Bicocca spin off Imaging and Vision Solutions, Member of European Laboratory for Learning and Intelligent Systems, and Member of the advisory board of the international AIQT Foundation, an international competence platform for the active public and private exchange of experience in the fields of artificial intelligence and quantum technology.