https://doi.org/10.2352/issn.2694-118X.2021.LIM-54
©2021 Society for Imaging Science and Technology

# Camera Colour Correction using Neural Networks

***Lindsay MacDonald, University College London, UK; Katarina Mayer, ESET, Bratislava, Slovakia***

## Abstract

*We demonstrate that a deep neural network can achieve near-perfect colour correction for the RGB signals from the sensors in a camera under a wide range of daylight illumination spectra. The network employs a fourth input signal representing the correlated colour temperature of the illumination. The network was trained entirely on synthetic spectra and applied to a set of RGB images derived from a hyperspectral image dataset under a range of daylight illumination with CCT from 2500K to 12500K. It produced an invariant output image as XYZ referenced to D65, with a mean colour error of approximately 1.0 $\Delta E^*_{ab}$.*

## Introduction

Colour constancy has been a long-standing research objective for digital camera technology. The ideal camera would adapt perfectly to any scene illumination, and produce signals dependent only on the surface reflectance at each point of the scene. The issue is that, in general, although the illumination source is unknown, it is intrinsic to the light reaching the camera, which is the product of the illumination spectrum and the surface reflectance spectrum at every point in the scene. If the illumination can be inferred or estimated then the scene colours can be quantified much more accurately.

The traditional method with trichromatic cameras was to perform a preliminary 'white balance' operation with a white card under the scene illumination, by adjusting the gain of each channel to produce equal signals. The same method can be applied post-capture by image processing if there is a white reference patch in the scene. Alternatively if the illumination source is known then a precalibrated white balance correction can be applied. The problem arises when the scene illumination is unknown and there is no available white reference.

Camera white balance depends on the sensor responses to the scene illumination. Once the colour of the illumination is known, a 3×3 diagonal matrix is applied to normalise the linear *RGB* triplets, analogous to the Von Kries correction of tristimulus response for human chromatic adaptation. The effect is to map the illumination tonal scale to the achromatic line in the camera's raw-*RGB* colour space, i.e. the values corresponding to the scene illumination are mapped to lie on the $R=G=B$ grey axis.

Most research into computational color constancy has been focused on illumination estimation, for which many methods have been proposed. Forsyth posed the colour constancy problem, given an image captured under an unknown illuminant, as that of determining the mappings which take the image gamut into the canonical gamut under a known illuminant.[1] A diagonal offset model with six parameters (three multipliers in a diagonal matrix plus three offsets) enables the gamut of colours to be estimated more accurately.[2] Instead of recovering a single estimate of the illuminant, image colours may be correlated with the sets of colours that can occur under each possible light source.[3] A second image may be taken through a suitable filter, yielding a multispectral image with six channels instead of three. The second (filtered) image may be deemed chromagenic if the relationship between the filtered and unfiltered *RGB* signals depends strongly on illumination.[4]

Neural networks have been widely applied to the problem of colour constancy. From a computational perspective, the goal of colour constancy can be defined as the transformation of a source image, taken under an unknown illuminant, to a target image, identical to one that would have been obtained by the same camera, for the same scene, under a standard illuminant. A multi-layer perceptron (MLP) was used to estimate the chromaticity of the illuminant in a scene based on the image data collected by a digital camera. The neural network was trained to learn the relationship between pixels in the scene and the chromaticity of the scene's illumination.[5] Zhan *et al* treated colour constancy as an illumination classification problem by training convolutional neural networks (CNNs). Input images were first clustered into groups of similar illumination with the K-means algorithm.[6]

We have previously shown that a neural network can perform better than polynomial regression for camera characterisation, i.e. the transformation of *R,G,B* sensor signals into *X,Y,Z* colorimetric stimuli [7]. The measured spectral sensitivities of a Nikon D200 camera were used for various network architectures, with both linear and sigmoidal activations. Training data was computed from a very large set of synthetic reflectance spectra, and test data computed from a compilation of real (measured) reflectance spectra, in all cases using the standard D65 illuminant. A neural network trained with the $\Delta E_{2000}$ error metric achieved a mean error of 0.89 $\Delta E^*_{ab}$ and more than 68% of all test samples had colour errors of less than 1.0 $\Delta E^*_{ab}$. This was significantly more accurate than could be achieved by linear regression over the same datasets, with a mean error of 1.83 $\Delta E^*_{ab}$ and 47% of samples having colour errors of less than 1.0 $\Delta E^*_{ab}$.

This study investigates how well a deep neural network can achieve colour correction of *RGB* signals from the sensors in a camera under a wide range of daylight illumination spectra. The network is trained not on images but on a million synthetic reflectance spectra whose colour gamut exceeds that of real materials. The trained network is applied to a set of *RGB* images derived from a hyperspectral image dataset under a range of daylight illumination with CCT from 2500K to 12500K.

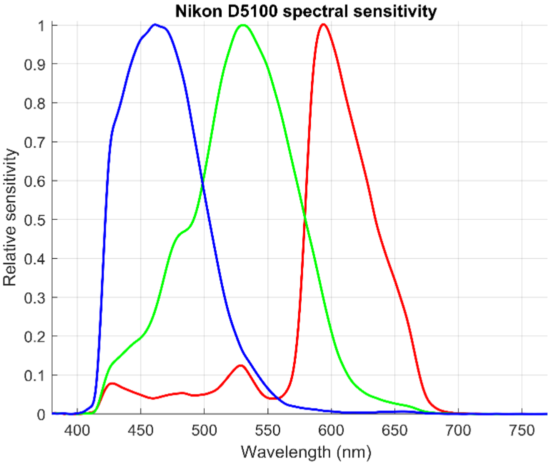

***Figure 1.*** *Normalised spectral sensitivities of Nikon D5100 camera*

## Method

The camera's *RGB* response is determined by the relative sensitivity of each channel for all wavelengths across the visible spectrum. We chose a Nikon D5100 camera with the spectral sensitivities measured by NPL [8] at intervals of 5nm over the range [380,780] nm, as shown in Fig. 1. For computation in Matlab, all spectral data was interpolated to 1nm intervals.

A training dataset of one million synthetic reflectance spectra was constructed from ten different basis functions, including gaussian, logistic, piece-wise linear ramp, sum of sines, and random walk, all with randomised parameter values. Each generated spectrum was constrained to have characteristics similar to the reflectance spectra of real surfaces, i.e. single-valued, continuous, limit on maximum slope, and conformance of first derivatives. Three independent datasets were used for testing network performance, all based on reflectance spectra measured from real surfaces: collated spectrophotometric measurements including several colour atlases; pigment spectra in the ISO 16066 (SOCS) standard [9]; and sampled pixel spectra from a set of hyperspectral images [10].

Using the discrete form of the colorimetric equation, the observer response $Y_n$ was calculated as the summed triple product (and similarly for $X_n$ and $Z_n$):

$$Y_n = \sum_{380}^{780} S(\lambda).R_n(\lambda).Y(\lambda).\Delta\lambda \qquad (1)$$

where: $Y_n$ is the observer response for sample $n$

$S(\lambda)$ is the power of the illuminant (D65)
$R_n(\lambda)$ is the reflectance factor for sample $n$ ( [0,1])
$Y(\lambda)$ is the CIE colour matching function
$\lambda$ is the wavelength (nm)
$\Delta\lambda$ is the wavelength interval = 1nm

*R,G,B* camera signals were calculated in the same way, using the sensitivity functions of Fig. 1. A 3→3 neural network (Fig. 2) was trained using the synthetic spectra, with a second independent dataset of 1M synthetic spectra as a validation set. The performance of the network was tested with each of the three datasets of measured reflectances. The network transformed *RGB*→*XYZ*, where the *RGB* triplets were calculated for the camera spectral sensitivities and the *XYZ* triplets were calculated for the CIE 1931 2° Standard Observer, both under the D65 illuminant.

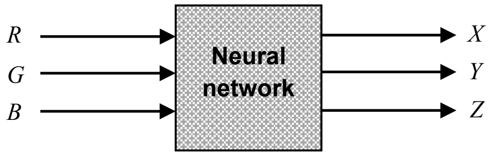

***Figure 2.*** *Neural network transforms camera RGB to colorimetric XYZ*

As a performance baseline, regression fitting was first employed to determine linear transforms from *RGB* to X*YZ*. Both sets of values were normalised, with the D65 white references of [*R*w,*G*w,*B*w] = [0.5814, 1.0, 0.8526] and [*X*w,*Y*w,*Z*w] = [ 0.9504, 1.0, 1.0886 ]. The equation is formulated as:

$$[\,R\;G\;B\,]\,\mathbf{M} = [\,X\;Y\;Z\,] \qquad (2)$$

where $R, G, B$ and $X, Y, Z$ are triple column vectors with $n$ rows (observations). The best-fitting 3×3 matrix **M** was found to be:

$$\mathbf{M} = \begin{bmatrix} 1.2074 & 0.4907 & 0.1790 \\ 0.1806 & 0.9680 & -0.3963 \\ 0.0745 & -0.3002 & 1.6110 \end{bmatrix} \qquad (3)$$

The first set of 1M synthetic spectra was employed for training, the second set of 1M synthetic spectra for validation, and the real spectra for testing. The error distributions always have long tails, with small errors for the majority of samples and very large error values for a small number of samples, resulting in the mean value being approximately double the median and the maximum value being over 10 times the mean. Colour differences were calculated as $\Delta E^*_{ab}$, between $L^*a^*b^*$ corresponding to the true *XYZ* value for each sample in the test dataset and the $L^*a^*b^*$ corresponding to the *XYZ* value predicted by applying the matrix to the *RGB* camera signals for the corresponding sample.

The results for linear matrix fitting of data under the D65 illuminant are given in Table 1. Values in the last column (denoted %E1) are the percentages of errors below the nominal perceptual threshold of $\Delta E^*_{ab}$=1.0.

**Table 1: Error ($\Delta E^*_{ab}$) statistics for regression fitting**

| Test dataset | # samples | med | mean | max | %E1 |
|---|---|---|---|---|---|
| Validation | 1,000,000 | 1.17 | 1.99 | 86.08 | 44.8 |
| Materials | 8,714 | 0.85 | 1.72 | 27.13 | 54.5 |
| Pigments | 3,999 | 0.99 | 1.85 | 30.64 | 50.2 |
| Hyperspectral | 10,000 | 1.07 | 2.30 | 24.09 | 47.6 |

Note that the regression technique used here (mldivide in Matlab) seems to fit a separate model for each output *X,Y,Z*, possibly ignoring interactions between the outputs. A better technique would be Linear Multi-output Regression, which takes into account possible relationships between the outputs, such as the covariance of *X* and *Y*. Another approach would be K-nearest-neighbours method, which is non-parametric and makes no assumptions about a functional relationship between inputs and outputs. It uses a given number of 'neighbouring' observations and makes predictions based on their average. (The K value of 5 seems to give a balanced bias-variance trade-off in this case.) Another non-parametric approach would be Decision Tree, which can capture nonlinear relationships between inputs and outputs. It learns by splitting the data into groups so that each group is as homogenous (similar), given the predicted variable, as possible.

## Neural Network

We experimented with many different model architectures for the neural network before finally selecting the best model. We employed the empirical Grid Search approach and explored all combinations of the parameters listed in Table 2. The model we selected has the following architecture: 3 inputs (*R,G,B*), 79 nodes in the first hidden layer, 36 nodes in the second hidden layer, and 3 nodes in the output layer (*X,Y,Z*).

**Table 2: Parameter values used in Grid Search procedure**

| | |
|---|---|
| Activation function | Relu, Sigmoid, Elu, Linear |
| Loss function | MSE, $\Delta E^*_{ab}$, ΔECIEDE2000 |
| Optimizer | Adam, Adagrad, RMSprop, SGD |
| # of layers | One, two, three |
| # of nodes in (hidden) layer | Layer 1: 10, 30, 50, 70, 90<br>Layer 2: 10, 30, 50, 70, 90<br>Layer 3: 10, 30, 50, 70, 90 |
| Learning rate | 0.001, 0.003, 0.005, 0.01, 0.03, 0.05 |
| # of training epochs | 15, 25, 35, 45, 55, 65, 75 |

The activation functions in both hidden layers are exponential linear units (Elu) and Linear in the output layer. We used the Adam optimiser and have allowed our model to train for up to 65 epochs but used the 'early stopping' technique, so the model stopped training once its performance longer improved on a hold-out validation dataset. This technique helps to prevent model overfitting and improves how the model generalizes to previously unseen data.[11] We also used the callback method during training, so that after each training epoch only the best model so far was saved.

Thus we ended up with the best model having two hidden layers. Models with only one hidden layer could not compete with models with two hidden layers, and models with three hidden layers did not generalise well, as they simply memorised the training data. The best performance was achieved with the CIEDE2000 loss function, a learning rate of 0.001, and 65 training epochs. In the second stage, we chose the optimal number of nodes selected for the two hidden layers (70 and 30) and ran another grid search procedure to fine-tune these numbers, by allowing the number of nodes in the first hidden layer to vary from 60 to 80 and in the second from 20 to 40. As a result, we chose the optimum configuration of 3 → 79 → 36 → 3 nodes.

The activations were Elu (exponential linear unit) for the second and third layers and Linear for the last. Elu is a type of activation function similar to Relu, but unlike Relu it allows small negative values to persist during data transformation [12]. The results are given in Table 3, with %errors < 1.0 and 2.0 $\Delta E^*_{ab}$.

**Table 3: Error ($\Delta E^*_{ab}$) statistics for 3→3 neural network**

| Test dataset | med | mean | 95%ile | max | %E1 | %E2 |
|---|---|---|---|---|---|---|
| Validation | 0.98 | 1.62 | 5.27 | 38.70 | 50.5 | 72.6 |
| Real materials | 0.84 | 1.42 | 4.56 | 11.59 | 56.1 | 77.0 |
| SOCS pigments | 0.94 | 1.43 | 4.38 | 12.90 | 52.3 | 77.2 |
| Hyperspectral | 1.08 | 1.58 | 4.75 | 8.74 | 47.3 | 72.2 |

The median errors produced by the network are similar to those of the polynomial, but the tail of the error distribution is shorter, because the network adapts to the non-linearity of the transformation. The network performance for the real materials is very similar to that for the SOCS pigments, but is somewhat worse for the hyperspectral data for two reasons: (a) the image spectra are noisier than spectrophotometric measurements, manifested as a jitter between adjacent wavelengths; and (b) for vegetation they rise sharply above 680 nm, the so-called 'red edge' caused by chlorophyll.

## Variable daylight illumination

A family of daylight illumination spectra was constructed, using the CIE D-illuminant technique with a fundamental plus weighted sums of two spectral components, the coefficients being calculated as functions of correlated colour temperature (CCT) [13]. A series of 10001 spectra was computed for CCT increments of one degree (1K) over the range [2500,12500] K, each at wavelength intervals of 1nm over the range [380,780] nm.

*RGB* triplets were calculated as before (Eq. 1), from the same synthetic reflectance spectra, but choosing the illumination spectrum at random from the daylight set. The corresponding *XYZ* triplets were calculated with the same sample reflectance and an invariant D65 illuminant. The objective is to use a 4→3 network to transform *RGB* under variable daylight to colour-constant *XYZ* under D65. The network (Fig. 3) transforms *RGBT*→*XYZ*, where *T* is a parameter representing CCT, scaled to the range [0,1], defined by Eq. 4.

$$T = \frac{CCT - 2500}{10000} \qquad (4)$$

As before, the *XYZ* output signals are always referred to D65 illuminant for the 2° observer.

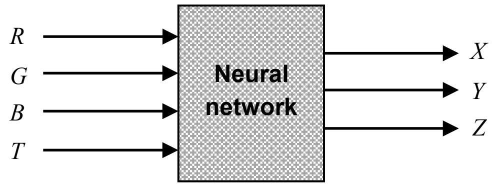

*Figure 3. Neural network RGB to colorimetric XYZ, with input parameter T*

A matrix was fitted to the data by linear regression to convert the *R,G,B,T* signals to *X,Y,Z* tristimulus values, formulated as:

$$[\,R\ G\ B\ T\,]\,\mathbf{M} = [\,X\ Y\ Z\,] \qquad (5)$$

where $R, G, B, T$ and $X, Y, Z$ are column vectors with $n$ rows (observations). The best fitting 4×3 matrix **M** was found to be:

$$\mathbf{M} = \begin{bmatrix} 0.9252 & 0.3910 & 0.6854 \\ 0.3144 & 1.0163 & -0.2125 \\ 0.0486 & -0.3133 & 1.0705 \\ 0.0212 & 0.0043 & -0.1115 \end{bmatrix} \qquad (6)$$

As before, the first set of 1M synthetic spectra was employed for training, the second set of 1M synthetic spectra for validation, and the real spectra for testing, with results in Table 4. Compared with Table 1, the errors are very disappointing, larger by a factor of more than 10, and it is clear that this is not a viable approach.

**Table 4: Error ($\Delta E^*_{ab}$) statistics for linear regression fitting with 4x3 matrix**

| Test dataset | med | mean | 95%ile | max | %E1 | %E2 |
|---|---|---|---|---|---|---|
| Validation | 10.41 | 16.30 | 52.74 | 173.44 | 0.8 | 3.6 |
| Real materials | 12.48 | 16.77 | 45.37 | 155.03 | 0.5 | 2.2 |
| SOCS pigments | 14.33 | 21.57 | 70.72 | 159.89 | 0.4 | 2.0 |
| Hyperspectral | 13.04 | 27.42 | 106.17 | 173.84 | 0.8 | 3.3 |

When a 4→3 neural network was trained on the same data, however, it gave the results in Table 5 which are excellent. The median errors are only about 20% larger than those of the 3→3 network given in Table 3.

**Table 5: Error ($\Delta E^*_{ab}$) statistics for 4→3 neural network**

| Test dataset | med | mean | 95%ile | max | %E1 | %E2 |
|---|---|---|---|---|---|---|
| Validation | 1.08 | 1.93 | 6.63 | 62.97 | 47.3 | 70.5 |
| Real materials | 0.94 | 1.61 | 5.12 | 17.72 | 52.2 | 74.8 |
| SOCS pigments | 1.06 | 1.69 | 5.16 | 18.90 | 48.3 | 73.0 |
| Hyperspectral | 1.21 | 1.85 | 5.42 | 12.14 | 43.5 | 66.2 |

This is an important result, because it shows that a neural network trained entirely on samples derived from synthetic spectra can achieve near-constant colour correction by converting from any daylight illumination to a fixed reference illumination.

## Performance on Hyperspectral Images

A series of ten *RGB* images was computed from a hyperspectral scene reflectance array [10], using the Nikon D5100 camera sensitivities and daylight spectra of ten CCT values (2500, 3000, 3500, 4000, 4500, 5500, 6500, 7500, 9500, 12500 K).

The resulting sequence of *RGB* ('camera raw') images for one scene is shown in Fig. 4. The image size is 819 x 812 pixels. Each pixel was initially represented by a vector of 31 reflectance values at intervals of 10 nm over the range [410,710] nm. These were extended to the range [380,780] nm and then interpolated to

1nm intervals, giving a vector of 401 values, before multiplying by illumination and observer/camera spectra.

Two different methods of white balance were compared: *RGB* camera signal correction and 4→3 neural network. The *RGB* correction method scales the camera *R,G,B* signals by a multiplicative factor for each CCT. It is equivalent to 'white card calibration', adjusting the values of white $R_w G_w B_w$ for the given daylight spectrum to what they would be under a D65 spectrum:

$$R' = \frac{R_{65}}{R_w} R\,, \qquad G' = \frac{G_{65}}{G_w} G\,, \qquad B' = \frac{B_{65}}{B_w} B \tag{7}$$

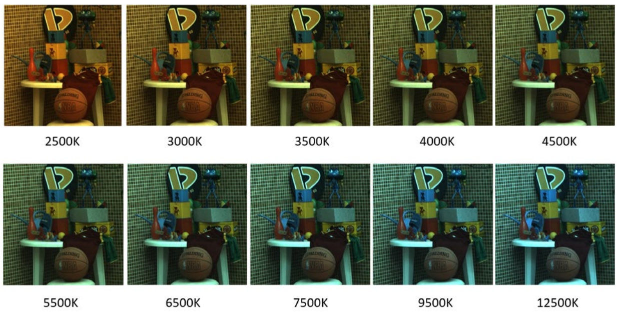

***Figure 4**. Ten 'camera raw' images computed from a hyperspectral scene with a series of daylight illuminants having CCT from 2500K to 12500K*

The white $R_w G_w B_w$ values were computed for each one degree (1K) increment of CCT. The transformed *XYZ* image was compared with the reference *XYZ* image under D65, both converted to $L^*a^*b^*$. The neural network gave better results for all CCT up to 10,000K than the *RGB* white balance, which is 'U-shaped', with a minimum at the reference CCT of 6500K. The statistics are quantified in Table 6 for differences between an image taken under daylight illumination of 4500K and the reference image under 6500K.

**Table 6: Error (ΔE*$_{ab}$) statistics for two methods of transforming a 4500K image to 6500K image**

| Method | med | mean | 95%ile | max | %E1 | %E2 |
|---|---|---|---|---|---|---|
| RGB white bal | 1.51 | 2.40 | 7.70 | 35.39 | 24.2 | 65.5 |
| Neural network | 1.09 | 1.55 | 4.28 | 18.05 | 46.3 | 74.5 |

Difference images for the two methods are shown in Fig. 5, where the $\Delta E^*_{ab}$ colour difference is shown as a scalar value in false colour, with an overall scale from 0 to 25 $\Delta E^*_{ab}$. The green sleeve of the sports shirt has significant errors in both cases; and the blue cube is substantially in error for the *RGB* white balance method. The latter also shows very large errors for the yellow of the Weetabix tin.

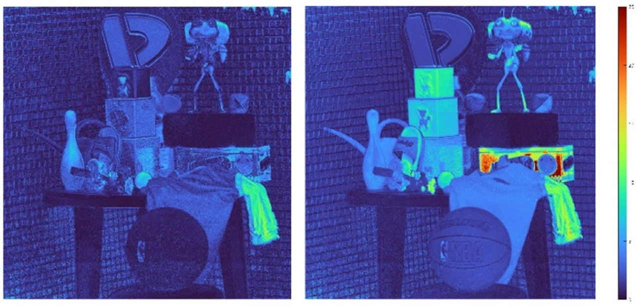

***Figure 5**. Differences between 4500K and 6500K images, showing regions of error: (left) neural network (right) RGB white balance method*

## Conclusion

In this computational study we have demonstrated that a 4→3 multi-layer neural network, trained entirely with samples derived from synthetic spectra, can achieve significantly better overall colour correction, quantified in terms of colorimetric accuracy, when transforming camera *RGB* signals to *XYZ* tristimulus values, than the traditional *RGB* 'diagonal matrix' approach to camera white balance. Moreover the neural network is effective over a wide range of daylight illumination spectra, with correlated colour temperatures from 2500K to 12500K. Approximately 50% of all colours of real materials give colorimetric errors of less than 1.0 $\Delta E^*_{ab}$ unit, and 75% give errors less than 2.0 $\Delta E^*_{ab}$ units. The effectiveness of the trained network was demonstrated on a hyperspectral test image.

In addition to the *RGB* input signals, the fourth input signal to the network, *T*, is a value representing the colour temperature of the illumination. This could be derived from external measurement of CCT, for example by a separate instrument, or directly from the camera itself by a calibration step with capture of a white card, or by including a known white target in the scene for post-capture image processing. It could also be estimated by analysis of the scene content.

The use of one additional signal *T* to index daylight was successful in this case because the family of daylight spectra is based on a model with single parameter (CCT). More generally, for an arbitrary illumination spectrum, one or more additional parameters would be needed.

## References

[1] D.A. Forsyth, A Novel Algorithm for Colour Constancy. Intl. J. Computer Vision, 5(1), 5-36 (1990)

[2] G.D. Finlayson, S.D. Hordley, and R. Xu, Convex programming colour constancy with a diagonal-offset model. Proc. IEEE Intl. Conf. on Image Processing Vol. 3, p.948 (2005)

[3] G.D. Finlayson, S.D. Hordley, and P.M. Hubel, Color by correlation: A simple, unifying framework for color constancy. IEEE Trans. Patt. Analysis and Machine Intelligence, 23(11), 1209-1221 (2001)

[4] G.D. Finlayson, S.D. Hordley, and P. Morovic, Colour constancy using the chromagenic constraint. Proc IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05) Vol. 1, pp.1079-1086 (2005)

[5] V.C. Cardei, A neural network approach to colour constancy. PhD Thesis, Simon Fraser University (2000)

[6] H. Zhan, S. Shi and Y. Huo, Computational colour constancy based on convolutional neural networks with cross-level architecture. IET Image Processing, 13(8), 1304-1313 (2019)

[7] L.W. MacDonald and K. Mayer, Color Space Transformation using Neural Networks. Proc. Color and Imaging Conf, 153-158 (2019)

[8] M.M. Darrodi, G. Finlayson, T. Goodman and M. Mackiewicz, Reference data set for camera spectral sensitivity estimation. JOSA A, 32(3), 381-391 (2015) The data was downloaded from https://spectralestimation.wordpress.com/data/

[9] Graphic technology — Standard object colour spectra database for colour reproduction evaluation (SOCS), CIE Technical Report ISO/TR 16066:2003, Vienna, Austria

[10] S.M. Nascimento, F.P. Ferreira and D.H. Foster, Statistics of spatial cone-excitation ratios in natural scenes. JOSA A, 19(8), 1484-1490 (2002)

[11] R. Caruana, S. Lawrence and L. Giles, Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in neural information processing systems (2001) 402-408

[12] D-A Clevert, T. Unterthiner and S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs). arXiv:1511.07289 (2015)

[13] R.W.G. Hunt and M.R. Pointer, Measuring Colour, 4th Ed., John Wiley, Chichester, p.374 (2011)