# A trainable Gaussian color model for determining the color invariant


Gianpaolo Gioia
g.g.gioia@student.tudelft.nl
TUDelft

Supervisors:
A. Lengyel (a.lengyel@tudelft.nl)
J.C. van Gemert (j.c.vangemert@tudelft.nl)


## 1. Introduction

- Improving robustness for day-night illumination shift.
- Incorporating physics based prior knowledge in form of color invariant edge detectors.
- Robustness important for autonomous driving and other safety-critical computer vision applications.

## 2. Background

By incorporating the color invariant in a Convolutional Neural Network (CNN), the performance of the CNN can be improved when dealing with an illumination shift [1]. The color invariant is applied to the input in the Color invariant Convolution (CIConv) layer to create a color invariant representation (Fig. 1) before used to train the CNN.
To calculate the color invariant, we need the spectral differential quotients, E, Eλ and Eλλ as is shown in

$$W \quad \begin{aligned} W &= \sqrt{W_x^2 + W_{\lambda x}^2 + W_{\lambda\lambda x}^2 + W_y^2 + W_{\lambda y}^2 + W_{\lambda\lambda y}^2}, \\ W_x &= \frac{E_x}{E}, \quad W_{\lambda x} = \frac{E_{\lambda x}}{E}, \quad W_{\lambda\lambda x} = \frac{E_{\lambda\lambda x}}{E} \end{aligned} \qquad (1)$$

every pixel is converted to the Gaussian color model with the following linear transformation:

$$\begin{bmatrix} \hat{E} \\ \hat{E}_\lambda \\ \hat{E}_{\lambda\lambda} \end{bmatrix} = \begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (2)$$

This matrix (1) is a product of the two matrices. With the first one approximating the XYZ basis, being:

$$\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} = \begin{pmatrix} 0.621 & 0.113 & 0.194 \\ 0.297 & 0.563 & 0.049 \\ -0.009 & 0.027 & 1.105 \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (3)$$

These matrices are fixed in the original CIConv layer and are an approximation for the conversion which creates potential for improvement.

## 3. Research question

*Can the performance of the CIConv layer be improved by making the transition to the Gaussian color model learnable?*

## 4. Method

**Baseline.** The fixed linear transformation (2) described in the background is an approximation for the conversion to the Gaussian color model and represents the baseline in the experiment.

1. **Linear learning.** The first set of experiments uses the fixed matrix as a basis and is automatically modified each epoch.
2. **Non-linear learning.** The matrix is replaced by a neural network that accepts three inputs and also outputs three values, representing the RGB and spectral differential quotients respectively.

**Option for 1 & 2**

*a)* Half. Where only the first transformation from RGB to the XYZ basis (3) is trained.
*b)* Full. Where the full transformation is trained. (2)

**Option for 1.**

*a)* Unclamped. The values in the matrix have full freedom of change.
*b)* Clamped. The values in the matrix can only navigate between certain values.

**Options for 2.**

I. Size of the network where there is a distinction between:
   *a)* Short network. A network consisting of 3-10-10-10-3.
   *b)* Long network. A network consisting of 3-10-10-10-3-10-10-10-3.

II. Applied activation function in the network when transitioning from a layer with 3 perceptrons to a layer with 10. So two applications in a long network. All other transformations are ReLU transformations.
   *a)* Sigmoid
   *b)* Tanh
   *c)* ReLU

## 5. Results & Conclusion

*All tests are performed on the ShapeNet dataset with synthesized images with different lighting strengths.*

| Method | Options | | | Accuracy |
|---|---|---|---|---|
| | Half/Full | Clamped? | Clamp range | |
| Baseline | N/A | N/A | N/A | 88.4 |
| Linear | Half | False | N/A | 85.2 |
| | | True | [-2.5, 2.5] | 89.0 |
| | | True | [-1.5, 1.5] | 87.5 |
| | Full | False | N/A | 88.2 |
| | | True | [-2.5, 2.5] | 88.5 |
| | | True | [-1.5, 1.5] | 88.9 |

*Table 1. Classification accuracy for linear learning variations.*

| Method | Options | | | Accuracy |
|---|---|---|---|---|
| | Half/Full | Short/Long | Activation | |
| Baseline | N/A | N/A | N/A | 88.4 |
| Neural Network | Half | Short | Sigmoid | 88.1 |
| | | | Tanh | **Failed** |
| | | | Relu | 86.7 |
| | | Long | Sigmoid | **Failed** |
| | | | Tanh | **Failed** |
| | | | Relu | **Failed** |
| | Full | Short | Sigmoid | 87.7 |
| | | | Tanh | **Failed** |
| | | | Relu | **Failed** |
| | | Long | Sigmoid | **Failed** |
| | | | Tanh | **Failed** |
| | | | Relu | **Failed** |

*Table 2. Classification accuracy for neural network variations.*

- The results in table 1 show that the when the values in the matrix are clamped, the best results in linear learning are achieved.
- The only consistent performing activation function is the sigmoid, showing potential to outperform the baseline when used in a network of most suitable size.
- We can conclude that making the transformation to the Gaussian color model can indeed improve the performance of the CIConv layer.
- Directions for future work include analysing the color invariant representations of the learned conversions and working with more sizes for the neural network. Also, a convolutional approach for the neural network can be explored where the neural network takes an area into account.

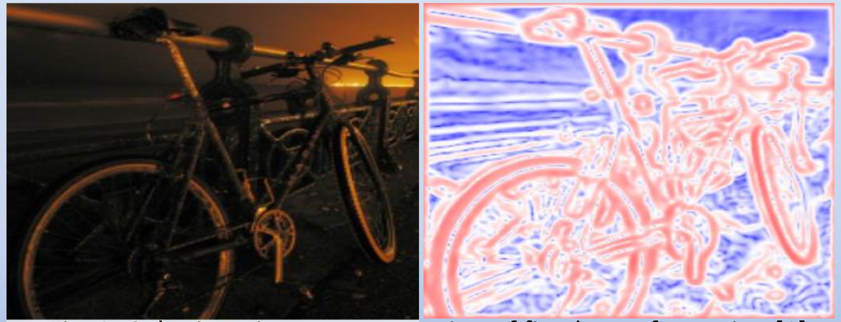

Fig. 1: Color invariant representation of fixed transformation [1]

## References


[1] Attila Lengyel, Sourav Garg, Michael Milford, and Jan C. van Gemert. Zero-shot domain adaptation with a physics prior. CoRR, abs/2108.05137, 2021.