A trainable Gaussian color model for determining the color invariant

Gianpaolo Gioia[1]
Supervisor(s): A. Lengyel[1] **, J.C. van Gemert**[1]
[1]EEMCS, Delft University of Technology, The Netherlands
g.g.gioia@student.tudelft.nl, {a.lengyel, j.c.vangemert}@tudelft.nl

Abstract

The possibility to improve an existing method by making (part of) it learnable is explored in this research. The work that this research extends added prior knowledge to a Convolutional Neural Network (CNN) to improve its performance when dealing with an illumination shift. The method used for the preprocessing, is the color invariant. The method was used in a zero-domain adaptation setting, where the network is trained without having access to the target domain. The research demonstrated improved performance, motivating further improvements.

The Color Invariant Convolution (CIConv) layer implements the color invariant edge detectors. The layer converts the RGB input of each pixel to spectral differential quotients, which are used to determine the color invariant representation. This is done through two fixed linear transformations that only approximate these values. This indicates that an even better approximation can be obtained by making this transition learnable.

Two methods are used to make this transition learnable; a linear learning method and a non-linear learning method. The linear learning method uses the original transformation but allows for change and the non-linear method replaces the linear transform with a neural network. Both methods show potential for achieving better results than the fixed transformation, but only the linear learning method actually does perform better in a classification experiment. All experiments are done following the zero-shot day to night domain adaptation on a synthetic dataset.

1 Introduction

Driving is a complex task performed under ever changing conditions and subsequently many factors contribute to a collision. A large percentage of road injuries is attributed to human perceptual error. There is sound evidence that the ability to avoid collisions is impeded under night time lighting [16]. The decreased visibility on the road makes it harder to identify objects. Clearly, night time driving is an issue for humans, but how about self-driving cars? Well, it happens to be that the deep image recognition methods used in self-driving cars are also sensitive to illumination shifts [12]. In such safety-critical applications, robustness is essential so scientists continue to try and find methods to improve existing methods.

The first generation of methods in the computer vision field are now categorized as classical or traditional computer vision. In classical computer vision, the methods are based on a descriptive analysis. A descriptive analysis entails specifying a comprehensible mathematical model that describes the phenomenon that we wish to observe [15]. Eventually, classical computer vision was succeeded by machine learning where the descriptive analysis was replaced by a predictive analysis. In machine learning, the goal is the discovery of underlying rules and forming predictive models, while minimising the error between the actual and predicted outcome [15]. To pursue this, machine learning utilises a training framework, consisting of Artificial Neural Networks, that is fed with a large training set for which the outputs are known. First the networks expanded (Deep Learning[1]) and then evolved into Convolutional Neural Networks (CNNs)[2]. The development of CNNs has had an exceptional impact in the computer vision field, especially in the ability to recognize objects and feature learning [21].

To further improve the performance of CNNs the input data can modified. Several methods of preprocessing the input images have already been explored [13; 20]. The research which this paper extends also uses the method of adding prior knowledge as a visual inductive bias [12]. The preprocessing method used in the research is an approach used in classical computer vision called color invariance [5; 6]. Rather than just using the intensity, the color invariance has the ability to apply segmentation based on color, providing a broader class of discrimination between material boundaries. While this is a widely researched topic [8; 4; 18], the method has never been used in this fashion. The positive results in using color invariants gave the incentive to

[1]A subset of machine learning where the neural network consists of more than 3-4 layers

[2]CNNs distinguish themselves by combining different types of neural layers in the architecture. Convolutional layers, pooling layers and fully connected layers [21]

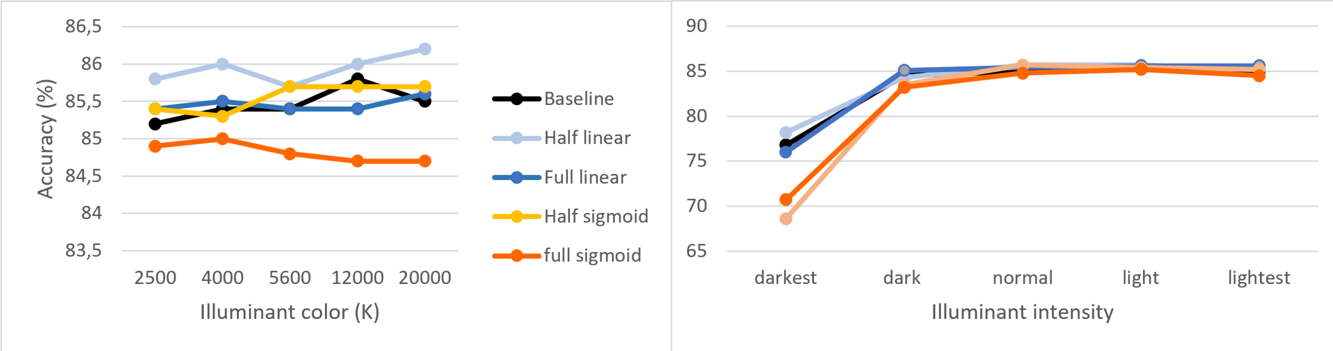

Figure 1: The classification accuracy of the best performing methods on the synthetic ShapeNet dataset. The half linear experiment is clamped in the range [-2.5, 2.5], the full linear is clamped in the range [-1.5, 1.5] and both best performing Sigmoid options use the defined short network. The tables with the results for the options are Table 1 and Table 3.

delve deeper in utilizing the physics-based reflection model as prior knowledge in a CNN. The color invariance reduces some of the complexity intrinsic to color images, with respect to known parameters of invariance, such as scene geometry, color, the intensity of the light source and the Fresnel reflection [5].

The work that has researched the color invariant as inductive basis uses a zero-shot domain adaptation [12]. This means that the network is trained without having access to the target domain. In this case is the network trained with day time data and tested against night time data to investigate the effectiveness of using the color invariant as part of a CNN. The experiments done in the prior research show an improved performance in this setting when using the color invariant contrary to using the non-preprocessed data in the same network [12]. This research will also use the same zero-shot domain adaptation in the experiments using the best performing color invariant W as baseline.

This research has focused on improving the layer in the network that applies the color invariant, the Color Invariant Convolution (CIConv) layer. Specifically, this paper will focus on improving the earlier fixed transformation to the Gaussian color model (GCM) [6] by making this part trainable. The Gaussian color model as described in the original setting is estimated by two linear transformations on the RGB camera responses. The first to estimate the XYZ basis and the second to estimate the spectral differential quotients used in determining the color invariant. Especially the linear transform to the XYZ basis is an approximate based on the camera sensitivities [2] and can vary based on the equipment used when recording the visuals [6]. The fact that the transition to the Gaussian color model is based on estimates indicates potential for improving the transformation and motivates to study the following research question:

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

To explore this research question, the fixed linear transformation to the GCM will be made into a learnable component using two techniques; a linear learning approach and a non-linear learning approach. The linear learning approach uses the original transformation but allows for changes according to the specified learning rate. The non-linear approach replaces the linear transform with a neural network, accepting three inputs and three outputs to match the initial conversion. Both methods are applied and tested to the whole transformation or the first part when changing to the XYZ basis. The results in Fig. 1 describe the best performing experiments for both methods under different lighting conditions. It shows that linear learning achieves the highest performance, outperforming the baseline. The half linear learning method performs better with the darkest intensity and the full linear learning method slightly better under dark conditions. This while the best performing non-linear experiments do not perform very well on the darker intensities.

The following contributions are part of this paper: (I) two general methods of making the GCM learnable are added to the CIConv layer, (II) different configurations of each method tested against a synthetic classification dataset and (III) a quantified evaluation of the experiments, determining which configurations are usable as a replacement of the fixed transformation.

2 Related Work

(Physics-Guided) Neural Networks

Nowadays, neural networks are a much used tool for finding underlying relations in a set of data. Although it has been proven a good method, in computer vision applications it was replaced by a better performing CNN, where LeNet started the era of CNNs [11]. CNNs typically have a standard structure where in contrast to traditional neural networks where all layers are fully connected layers, it combines 3 different kind of layers and only the last layer is fully connected [21]. Multiple works have suggested for CNNs to substitute networks with only fully connected layers with a view to attaining faster learning times (e.g. [14; 19]). The next development for the networks was adding prior knowledge from physical models to CNNs. The performance of these networks can significantly improve without adding more trainings data by adding the bias. For instance, adding translation equivariance through a convolutional prior. This has proven useful in applications such as line detection

[13] and spectral leakage [20]. Another way of adding prior knowledge is the usage of image formation models. This is the process of transforming the input image before putting it through a CNN. An examples is intrinsic image decomposition [1], that separates an image into its formation components such as reflectance and shading. The CIConv layer [12] also uses this technique and transforms the input to a color invariant representation. This work focuses on further developing the CIConv by making a fixed part of the color invariant learnable.

Color Invariant

The research on improving the invariance to illumination changes is a well-documented topic in the classical computer vision [8; 4; 18]. Early reflection models are derived from the Kubelka-Munk (KM) theory [10; 5]. This KM model uses four parameters of known invariance. The Scene geometry, that concerns the formation of shadows and shading, the color and illumination intensity and Fresnel reflections, that occurs on shiny objects that reflect light directly from the surface without interacting with the material color. Five relevant combinations derived from the model are determined [5] and tested when implemented in the first layer of a CNN [5]. The Color Invariant Convolution (CIConv) layer uses the edge detectors from [7] and shows the most promising test results when using the invariant that only has invariance to illumination intensity. This work focuses on the most successful invariant in the CIConv layer, leaving the other four invariant combinations of the KM model out of consideration. The CIConv layer of [12] is used while solely modifying the process of calculating Gaussian color model.

3 Method

The CIConv layer

The CIConv layer uses the edge detectors from [7]. These edge detectors are derived from the Kubelka-Munk theory [10], which is considered as a general model for color image formation. This photometric reflectance model describing the spectrum of light E reflected from an object is given by

$$E(\lambda, \vec{x}) = e(\lambda, \vec{x})((1-\rho_f(\vec{x}))^2 R_\infty(\lambda, \vec{x}) + \rho_f(\vec{x})), (1)$$

in which the $\vec{x}$ denotes the position in the image plane, and λ the wavelength. The $e(\lambda, x)$ stands for the illumination spectrum and $\rho_f(x)$ the Fresnel reflection at $\vec{x}$. While different combinations of invariance are possible derived from this model, in this use-case the W invariant proved to be the best performing invariant [12]. This invariant is only invariant to the illumination intensity. It assumes spectrally and spatially uniform illumination, so the $e(\lambda, \vec{x})$ can be represented by a constant i. It also assumes only matte surfaces making $\rho_f(\vec{x})) = 0$ reducing (1) to

$$E(\lambda, \vec{x}) = iR_\infty(\lambda, \vec{x}).(2)$$

The definition of the W invariant is given by

$$W = \sqrt{W_x^2 + W_{\lambda x}^2 + W_{\lambda\lambda x}^2 + W_y^2 + W_{\lambda y}^2 + W_{\lambda\lambda y}^2}, (3)$$

with the component defined as $W_x = \frac{E_x}{E}$, $W_{\lambda x} = \frac{E_{\lambda x}}{E}$ and $W_{\lambda\lambda x} = \frac{E_{\lambda\lambda x}}{E}$. To estimate the spectral differential quotients E, E_λ and $E_{\lambda\lambda}$, the Gaussian color model (GCM) [6] is used. The GCM converts the RGB responses with two linear transformations to the spectral differential quotients. A RGB-camera approximates the CIE 1931 XYZ basis for colorimetry by the linear transform [2]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{pmatrix} 0.62 & 0.11 & 0.19 \\ 0.3 & 0.56 & 0.05 \\ -0.01 & 0.03 & 1.11 \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.(4)$$

In [6] an approximate solution is also obtained to go from the XYZ basis to the Gaussian basis, defined by

$$\begin{bmatrix} E \\ E_\lambda \\ E_{\lambda\lambda} \end{bmatrix} = \begin{pmatrix} -0.48 & 1.2 & 0.28 \\ 0.48 & 0 & -0.4 \\ 1.18 & -1.3 & 0 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}.(5)$$

The product of Eq.(4) and Eq.(5) gives the desired conversion, specifying the GCM in RGB terms,

$$\begin{bmatrix} E \\ E_\lambda \\ E_{\lambda\lambda} \end{bmatrix} = \begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.035 \\ 0.34 & -0.6 & 0.17 \end{pmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}.(6)$$

To determine subsequently the spatial differential quotients E_x, $E_{\lambda x}$ and $E_{\lambda\lambda x}$ the spectral differential quotients are convolved with Gaussian derivative kernels g with standard deviation σ, e.g.

$$E_x(x, y, \sigma) = \sum_{t \in \mathbb{Z}} E(t, y) \frac{\partial g(x-t, \sigma)}{\partial x}.(7)$$

The parameter σ in Eq.(7) is the trainable parameter in the CIConv layer and defines the scale at which the image is convolved. This bigger σ is, the less details are preserved. By making this parameter trainable, it is possible to find a good trade-off between omitting noise and preserving details.

Gaussian color model

This research focuses on the one part of the CIConv layer, namely the transition to the Gaussian color model, used in determining the color invariant in the layer. As explained earlier, the original CIConv layer deals with this transformation with two fixed matrix multiplications, (4) and (5), which are approximations [6]. To try and optimize this process, the linear transforms will be turned into a trainable element. The first distinction in is between making the only (4) learnable and making the whole transform, identified as (6), learnable. The reason why making (5) on itself learnable is not researched, is because this transform from the XYZ basis to the spectral differential quotients is based on a reasoning equally applicable to all occurring circumstances, regardless of the recording conditions of the images. Whereas (4) is dependent on the camera sensitivities and can differ for each individual type of camera. Different approaches for the switch are explored and tested as part of the CIConv layer described in [12].

<table>
<tr><th>Method</th><th>Clamped?</th><th>Clamp range</th><th>Val acc(%)</th><th>Darkest acc(%)</th><th>Dark acc(%)</th><th>Normal acc(%)</th></tr>
<tr><td>Baseline</td><td>N/A</td><td>N/A</td><td>88.7</td><td>76.8</td><td>84.5</td><td>85.4</td></tr>
<tr><td rowspan="3">Half linear</td><td>False</td><td>N/A</td><td>88.7</td><td>77.6</td><td>84.3</td><td>85.3</td></tr>
<tr><td>True</td><td>[-1.5, 1.5]</td><td>87.5</td><td>77.9</td><td>84.3</td><td>85.3</td></tr>
<tr><td>True</td><td>[-2.5, 2.5]</td><td>89.0</td><td>78.2</td><td>84.3</td><td>85.7</td></tr>
<tr><td rowspan="3">Full linear</td><td>False</td><td>N/A</td><td>88.7</td><td>75.4</td><td>84.5</td><td>85.4</td></tr>
<tr><td>True</td><td>[-1.5, 1.5]</td><td>88.9</td><td>76.0</td><td>85.1</td><td>85.4</td></tr>
<tr><td>True</td><td>[-2.5, 2.5]</td><td>88.5</td><td>74.9</td><td>84.6</td><td>85.4</td></tr>
</table>

Table 1: ShapeNet classification accuracy of a ResNet-18 architecture with the linear learning method applied. Each experiment is performed three times with the best performing results displayed here.

For the generation of the learnable element, two methods are explored. The first is using the values of the existing linear transforms of (4) and (6) but allowing the learning rate of the network to alter the matrix. This is the same approach used for training the scale in the CIConv layer and uses the same learning rate. The second approach is replacing the matrix by a neural network that accepts three inputs and produces three outputs, representing the RGB and spectral differential quotients or XYZ basis respectively. In the neural network, different variations of the size of the network and the activation functions used, are experimented with.

4 Experiments

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

To answer this question, various classification experiments have been carried out with various settings applying the linear learning method or a non-linear method in the form of a neural network. These methods are compared to the performance of the initial fixed linear transformations used in the CIConv layer defined in [12], specified as *baseline* in the experiments. Both learning methods are applied to only the transformation described in (4) or the whole transformation of (6). From now on, these experiments are described as *half* or *full* learning respectively. To evaluate the performance of the methods not only the classification accuracy, but also the behavior of the learnable scale parameter and stability in training are taken into account.

Overall settings & dataset

The experiments on the datasets use a zero-shot domain adaptation approach [17; 12], which means that the network is trained without having access to the target domain, where in this case day and night time represent the training and target domain respectively.

All the experiments will be performed on images rendered from a subset of the ShapeNet [3] dataset. ShapeNet is a synthetic image dataset composed of images that are rendered in different illumination conditions. With accurate control of light intensity, the scene is point light modeled with temperatures ranging between [19000,20000]K and an ambient light source. The training set contains 1,000 samples for each of the 10 object classes recorded under ”normal” conditions (T=6500K). The test sets consist of 300 samples per class with a variety of light intensity and colors.

A baseline ResNet-18 [9] is trained in combination with the W color invariant in the CIConv layer. The training settings are the same as in [12]; training is done for 175 epochs with a batch size of 64 using SGD with momentum 0.9, weight decay 1e-4 and an initial learning rate of 0.05 with stepwise reduction by factor 0.1, step size 50. The experiment uses data augmentation in the form of random horizontal flips, random cropping and random rotation.

4.1 Linear learning

Can the performance of the CIConv layer be improved by applying a linear learning method to the values in the matrices used in the linear transform to the Gaussian color model?

The linear learning method uses the the matrix transformations of (4) and (6) as basis but allows for modifications according to the learning rate of the network. The scale is trained using the same method in the initial CIConv layer. The options for this experiment contain an unclamped option and clamped options containing the values of the matrices within the absolute values of 1.5 and 2.5. The restraining of the values in the half and full learning methods is applied to the matrices of the linear transforms of (4) and (6) because they already contain good approximations for the process and makes sure that the values do not explode. Thus, the options for the clamping are chosen in such a way to keep the values reasonably close to the initial values of the matrices.

The results in Table 1 show that both half and full have a better performing option than the baseline. For half learning is the best result achieved when clamped with the absolute value of 2.5 at an accuracy of 89.0% and for full learning is it achieved when clamped with the absolute value of 1.5 at 88.9%. The unclamped experiments even show a drop in performance, demonstrating that the clamping is vital for producing optimal results with this method.

The experiments show that the overall performance of the network can be increased by applying linear learning method to the matrices used in the transition to the Gaussian color model. In both the half and the full learning option, making (4) and (6) learnable respectively, do the experiments in Table **??** show increased accuracy compared to the baseline experiment. However, the learned matrices in Table 2 of these experiments do not resemble initial matrices anymore, suggesting that the new produced color invariant representations also do not resemble the initial invariant representation anymore. However, this research does not contain

Method	Clamp range	Initial matrix	learned matrix
Half learning	[-2.5, 2.5]	$\begin{pmatrix} 0.62 & 0.11 & 0.19 \\ 0.3 & 0.56 & 0.05 \\ -0.01 & 0.03 & 1.11 \end{pmatrix}$	$\begin{pmatrix} -1.86 & -1.86 & -2.43 \\ 2.49 & 2.09 & 2.19 \\ 2.50 & 2.42 & 2.35 \end{pmatrix}$
Full learning	[-1.5, 1.5]	$\begin{pmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.035 \\ 0.34 & -0.6 & 0.17 \end{pmatrix}$	$\begin{pmatrix} 1.06 & 0.32 & 0.26 \\ 1.45 & -0.40 & -1.47 \\ 1.17 & -1.24 & 0.10 \end{pmatrix}$

Table 2: The learned matrices of the best performing experiments displayed in Table 1. The initial matrices are the matrices of (4) and (6) described in the Method

.

the invariant representations of the experiments and only contains the quantifiable data of the experiments such as accuracy and scale, so no definite conclusions about the produced color invariant representation can be made.

4.2 Non-linear learning

Can the performance of the CIConv layer be improved by replacing the linear transform to the Gaussian color model by using a non-linear learning method?

The non-linear learning method uses a neural network to estimate the transformation to the GCM. The neural network that replaces the transformation accepts three inputs and has three outputs. The input will always correspond to the RGB, but the output can correspond to the XYZ basis or the spectral differential quotients depending on if half or full learning is applied. Two properties of the neural networks are explored in the experiments. The first is the activation function applied when the layer transitions to a layer with three perceptrons and the second is the size of the neural network. The second property is the size of the network where the following two sizes are used:

1. The *short* option that has three hidden layer of 10 perceptrons, described as $3-10-10-10-3$ and
2. the *long* option that has 7 hidden layer and applies the short option two times in succession described as $3-10-10-10-3-10-10-10-3$.

The standard activation function used in the layers in both structures is the Rectified Linear Unit (ReLU). It is used in every transition, except when the transition is made to a layer with three perceptrons. This means that the option for the activation function is either used once in the short option, or twice in the long option. The three most used activation functions are the options in the two specified network sizes. These three activation functions are the following:

1. The Sigmoid activation function, with formula
$$f(x) = \frac{1}{1+e^{-x}}$$
and range $[0, 1]$,
2. the Tanh activation function, with formula
$$f(x) = \frac{e^x - e^-x}{e^x + e^-x}$$
and range $[-1, 1]$ and
3. the ReLU activation function, with formula
$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{otherwise} \end{cases}$$
, and range $[0, >$.

Method	Size	Activation function	Accuracy
Baseline	N/A	N/A	88.7
Half linear	Short	Sigmoid	88.1
		Tanh	10.0
		ReLU	86.7
	Long	Sigmoid	10.0
		Tanh	10.0
		ReLU	10.0
Full linear	Short	Sigmoid	87.7
		Tanh	87.8
		ReLU	10.0
	Long	Sigmoid	10.0
		Tanh	10.0
		ReLU	10.0

Table 3: ShapeNet classification accuracy of a ResNet-18 architecture with the W invariant applied in the CIConv layer. All experiments are each performed three times with the best performing results displayed here.

The results in Table 3 show performances of the short and long options in combination with the three activation functions in the neural network. A first observation shows that the long network is not capable of learning the transition irrespective of which activation function is used. The short option does have successful experiments together with all three activation functions, with accuracies that come near the one of the baseline. So the composition of the layers is a key factor when using a neural network to determine the Gaussian color model. The network loses the capability to learn the transition when it has too many hidden layers.

Table 4 shows the stability of only the short network. Because of the unstable performances, the amount of successful experiments for each option is an important factor to consider. The TanH option was only successful 1 out of 6 experiments, the ReLU 2 out of 6, while the Sigmoid was successful in all experiments. This makes the Sigmoid the only stable activation function always giving proper results that get close to the performance of the baseline.

Method	Activation function	Successful experiments
Baseline	N/A	3/3
Half linear	Sigmoid	3/3
	Tanh	0/3
	ReLU	2/3
Full linear	Sigmoid	3/3
	Tanh	1/3
	ReLU	0/3

Table 4: Describing the amount of successful experiments to evaluate the stability of the activation functions in the short network. The only stable performing activation function is the Sigmoid function.

In the non-linear learning method, using a neural network no improved performances with respect to the baseline experiment can be observed. Nonetheless, no definite conclusions saying that this method can not improve the performance, can be made. The reason for the inconclusiveness is the instability when using a neural network. Although the Sigmoid activation function works very stable in the short network as described in Table 4, the failing of the long network indicates that the performance is rather sensitive to the explored properties of the size of the networks and activation function used. There is one tested configuration providing stable and decent results for both half and full learning. It is not performing better than the baseline, but because only one configuration in this method was stable, we can not conclude that this is the only and best working one. This means that this method still has the potential to do better considering there exists another unexplored composition of the network that optimizes the performance for this specific problem.

5 Discussion

The experiments for several options in both linear and non-linear methods show promising results.

The linear learning method shows improved performances with respect to the baseline experiments in half learning and full learning. These results imply that this method is readily usable in the CIConv layer for either option. However, the learned matrices in figure 2 do not resemble the initial matrices anymore. This suggests that applying the matrix multiplication of (5) to the learned matrix in half learning has become an unneeded calculation in this method. The goal of (5) is going from the XYZ basis to the Gaussian color model but the result of the learned linear transform does not approximate the transition to the XYZ basis anymore. So in future work, solely the full learning approach should be further explored. The produced color invariant representation can be evaluated to see how it differs from the initial one. Also more experiments with the clamping value can be carried out, without considering the initial values in the matrix.

The non-linear learning method does not show improved performance with respect to the baseline experiments in half and full learning. Still, the performance when applying the Sigmoid activation function shows potential to rise above the baseline experiment. Only when using this activation function, the performance of the network is stable and close to the accuracy of the baseline. No conclusions about the size of the network can be made based on the results of this experiment, because there was only one size that has produced decent results. Also, the only variation in the size is in the amount of hidden layers, keeping the amount of perceptrons in each hidden layer the same. So in future work the performance of more network compositions can be explored in term of the number of layers and perceptrons per layer.

An alternative non-linear learning method can also be considered in further experiments. Instead of letting a neural network consider the transformation per pixel, the setting can be changed to an approach more similar to a convolutional one. This means that not only the target pixel is considered in the transformation, but that the surrounding pixels are also taken into account in the determination of the Gaussian color model.

6 Responsible Research

The ultimate goal of the neural networks trained in this application, is to be applied in the recognition systems of self-driving cars. As this is a safety-critical application, the current experiments are not enough to validate the performance of these networks in real-life. The performance of the classification experiments described in the paper only demonstrate the potential for improving an already existing method that is also still in a development stage. The current implementation does not ensure a stable performance in the real world.

Reproducibility

The ShapeNet dataset used in the experiments is an open source dataset that is available for anyone to use. In combination with the available source code of this experiment, all experiments done when the instructions in the repository are followed. Not all configurations of the experiments are readily applicable, but following the instructions in 4.1 and 4.2 the parameters and implementation can be altered in such a way that all experiments can be reproduced.

References

[1] HG Barrow, JM Tenenbaum, AR Hanson, and EM Riseman. *Computer vision systems*. 1978.

[2] ITU-R Recommendation BT. Basic parameter values for the hdtv standard for the studio and for international programme exchange. 1990.

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[4] Ron Gershon, Allan D Jepson, and John K Tsotsos. Ambient illumination and the determination of material changes. *JOSA A*, 3(10):1700–1707, 1986.

[5] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350, 2001.

[6] Jan-Mark Geusebroek. Color and geometrical structure in images. *Appl. Microsc*, pages 116–119, 2000.

[7] Jan-Mark Geusebroek, Anuj Dev, Rein Van Den Boomgaard, Arnold WM Smeulders, Frans Cornelissen, and Hugo Geerts. Color invariant edge detection. In *International Conference on Scale-Space Theories in Computer Vision*, pages 459–464. Springer, 1999.

[8] Theo Gevers and Arnold WM Smeulders. Color-based object recognition. *Pattern recognition*, 32(3):453–464, 1999.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Paul Kubelka. Ein beitrag zur optik der farbanstriche (contribution to the optic of paint). *Zeitschrift fur technische Physik*, 12:593–601, 1931.

[11] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

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

[13] Yancong Lin, Silvia L Pintea, and Jan C van Gemert. Deep hough-transform line priors. In *European Conference on Computer Vision*, pages 323–340. Springer, 2020.

[14] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[15] Niall O'Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, and Joseph Walsh. Deep learning vs. traditional computer vision. In *Science and Information Conference*, pages 128–144. Springer, 2019.

[16] S Plainis, IJ Murray, and IG Pallikaris. Road traffic casualties: understanding the night-time death toll. *Injury Prevention*, 12(2):125–138, 2006.

[17] Eduardo Romera, Luis M Bergasa, Kailun Yang, Jose M Alvarez, and Rafael Barea. Bridging the day and night domain gap for semantic segmentation. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1312–1318. IEEE, 2019.

[18] Steven A Shafer. Using color to separate reflection components. Technical report, ROCHESTER UNIV NY DEPT OF COMPUTER SCIENCE, 1984.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[20] Nergis Tomen and Jan van Gemert. Spectral leakage and rethinking the kernel size in cnns. *arXiv preprint arXiv:2101.10143*, 2021.

[21] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.