Color Invariant Convolution for semantic segmentation

Nicky Ju[1]
Supervisor: Attila Lengyel[1] **Responsible Professor: Jan C. van Gemert**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

Abstract

Color Invariant Convolution (CIConv) is a learnable Convolutional Neural Network (CNN) layer that reduces the distribution shift between the source and target set in the CNN under an illumination-based domain shift [13]. We explore the semantic segmentation performance for day-night domain adaptation when using CIConv. We will test this on two settings: one with only labeled train data available and one with access to both labeled training data and unlabeled test data. In both settings, we will cast an invariant edge detector as a trainable CIConv layer in the CNN to transform the daytime dataset to a domain invariant representation. We will execute day-night domain adaptation and evaluate the mean Intersection over Union over the results. We compare this result to the vanilla version of the same code without using the invariant edge detector as a trainable layer. We will discuss the results obtained from our experiments and show that the trainable CIConv layer does not always result in better outcomes for day-night domain adaptation.

1 Introduction

Deep image recognition methods are sensitive to illumination shifts and illumination changes caused by, for example, day of time or weather [1; 7; 15]. Robustness to such recording conditions is essential for safety-critical computer vision applications, such as autonomous driving. However, adding extra test data is often challenging as it may be expensive and time-consuming to obtain. Furthermore, it would be impossible to collect training data for all possible scenarios in advance [13].

Semantic segmentation aims to assign a label to every pixel in an image. We can use a Convolutional Neural Network (CNN) to adapt semantic segmentation data to other domains, for instance, adapting from daytime data to nighttime data (see Figure 1). However, this trained model may not generalize well to unseen images, especially when there is a domain gap between the training (source) and test (target) images [19].

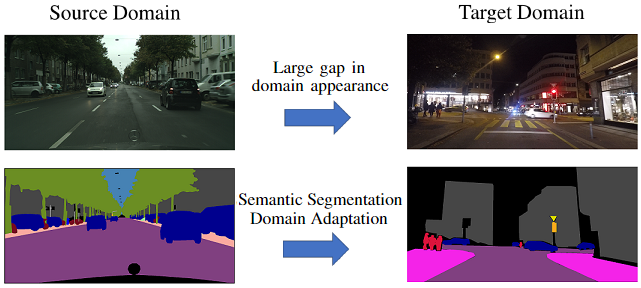


Figure 1: Representation of domain adaptation for a large domain gap in appearance.

Although color is often used to determine the distinction between objects, albeit a decisive clue, it may not always be the most efficient or coherent tool to use when adapting from domain. We can use color invariant edge detectors as these measure object properties independent of the imaging conditions derived from the measured color values [11].

In figure 1 you can see a simple representation of domain adaptation. Even if images from the two domains appear very different (especially in color), their segmentation outputs share a significant amount of similarities. Recent work has introduced Color Invariant Convolution (CIConv) [13], which incorporates color invariant edge detectors into a CNN to improve their robustness to day-night-related illumination changes without the need for any nighttime data.

In this paper, we will test and examine CIConv in two different settings. One setting which is used as a baseline uses a zero-shot setting. Another setting uses an approach called unsupervised domain adaptation. This setting performs feature learning, domain adaptation, and classifier learning jointly in a unified architecture, using a single learning algorithm (backpropagation). This can be trained on labeled data from the source domain and unlabeled data from the target domain [9].

This paper will evaluate the effectiveness of CIConv for unsupervised domain adaptation methods for semantic segmentation. We ask ourselves the following research question: **What is the effectiveness of CIConv for unsupervised domain adaptation for semantic segmentation?** With this research question, we ask ourselves the following sub-questions:

- What is the effect of CIConv for unsupervised domain adaptation compared to not using CIConv?
- How usable is semantic segmentation used on unsupervised domain adaptation with CIConv?

We have the following contributions: (i) we evaluate two different settings, zero-shot domain adaptation and an unsupervised domain adaptation; (ii) we evaluate these two different settings with and without a CIConv layer; (iii) we will show and discuss the results on the contributions above and will give elaborations on why the results turned out this way. All code will be made available on our GitLab page.[1]

2 Related Works

Color Invariants Although color is an often used attribute to use in the distinction between objects [11], physics-based reflection models to improve invariance to illumination changes is also a well-researched topic in computer vision [4]. We consider the determination of material changes independent of the illumination color and intensity. Meaning we lose the features that we would gain from the color illumination but gain features based on material changes [11]. Early work includes invariants derived from the Kubelka-Munk reflection model [10; 16]. Since then, methods have been proposed for shadow removal or intrinsic image decomposition with (street) image segmentation applications [20]. Recent works have shown improved segmentation performance by applying a color invariant transformation as a preprocessing step [2; 3]. Our work further explores the results of using a classical color invariant as a trainable CNN layer.

Semantic Segmentation Semantic segmentation is a crucial component in image understanding. The goal of semantic segmentation is to assign a unique label (or category) to every single pixel in the image, which can be considered a dense classification problem [14]. The interpretations of these label images can then be used for many critical applications, such as autonomous driving, robotic navigation, localization, and scene understanding [21]. This paper will use two different architectures of semantic segmentation: RefineNet and Deeplab.

Refinenet is a generic refinement model that makes use of all the information available along the down-sampling process to enable high-resolution prediction using long-range residual connections. The individual components of RefineNet employ residual connections following the identity mapping mindset, which allows for effective training [14].

Deeplab is a semantic segmentation model that achieves prediction by up-sampling the output of the last convolution layer and computing pixel-wise loss [17]. The Deeplab applies atrous spatial pyramid pooling (ASPP) for up-sampling. ASPP examines an incoming CNN feature layer with filters at multiple sampling rates and effective fields-of-views. Therefore, ASPP captures objects and image context at various scales [5].

[1] https://gitlab.tudelft.nl/attilalengyel/brp-ciconv/-/tree/master/Nicky%20Ju

Unsupervised Domain Adaptation Learning classifiers and features in the presence of a shift between training and test distributions is known as domain adaptation [9]. However, this classifier or feature may not adapt well to unseen images, especially when there is a domain gap between the training (source) and test (target) images. That is where unsupervised domain adaptation will play a relevant role. When there are labeled train data and non-labeled test data available, the unsupervised domain adaptation will give labeled test data as a result. This is important because relying on the supervised model where every pixel of an image requires manually classifying by a human would entail prohibitively high labor costs. This approach requires no additional data sources and thus avoids expensive data gathering costs.

3 Method

Our color invariant layers make use of an invariant edge detector from [10]. A CIConv layer is a learnable color invariant CNN layer that reduces the activation distribution shift in a CNN under an illumination-based domain shift such as the domain shift of going from daytime to nighttime [13]. The CIConv can be used with different kinds of color invariants, where some invariants yield better results than others. In this paper, we will only work with one specific invariant:

$$\begin{aligned} W &= \sqrt{W_x^2 + W_{\lambda x}^2 + W_{\lambda\lambda x}^2 + W_y^2 + W_{\lambda y}^2 + W_{\lambda\lambda y}^2} \\ W_x &= \tfrac{E_x}{E}, W_{\lambda x} = \tfrac{E_{\lambda x}}{E}, W_{\lambda\lambda x} = \tfrac{E_{\lambda\lambda x}}{E} \end{aligned} \tag{1}$$

The Gaussian color model [24] is used to estimate E, E_λ and $E_{\lambda\lambda}$ from the RGB camera responses as

$$\begin{bmatrix} E(x,y) \\ E_\lambda(x,y) \\ E_{\lambda\lambda}(x,y) \end{bmatrix} = \begin{bmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{bmatrix} \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix} \tag{2}$$

where x, y are the pixel locations in the image. Spatial derivatives E_x and E_y are calculated by convolving E with a Gaussian derivative kernel g with standard deviation σ, i.e.

$$E_x(x,y,\sigma) = \textstyle\sum_{t\in Z} E(t,y) \frac{\partial g(x-t,\sigma)}{\partial x} \tag{3}$$

and similarly for E_y, $E_{\lambda x}$, $E_{\lambda\lambda x}$, $E_{\lambda y}$ and $E_{\lambda\lambda y}$.

CIConv is described as:

$$CIConv(x,y) = \frac{log(CI^2(x,y,\sigma=2^s)+\epsilon) - \mu_S}{\sigma_S} \tag{4}$$

with CI the color invariant W from equation 1, μ_S and σ_S the sample mean and standard deviation over log $(CI^2 + \epsilon)$, and ϵ a small term added for numerical stability [13].

In [13] it has been tested and verified that this invariant works best based on mIoU compared to other invariants and that is the reason why this invariant will be the only invariant further explored and tested in this paper.

Given the source data set with segmentation labels and the target data set with no labels, we want to train a network for semantic segmentation, which is finally tested on the same target data set.

We decided to test this with a zero-shot setting as a baseline. This [13] code base will be used for this, with CIConv implemented in one run, and without CIConv implemented in a different run. This will be compared to an unsupervised domain adaptation setting. AdaptSegNet [19] will be used for this; one run with CIConv implemented and one without CIConv implemented.

For determining the performance, we will utilize the Intersection over Union (IoU) score. This approach is significantly better than pixel accuracy since our images often have a problem called class imbalance, which means that a portion of our classes dominate the image while some other classes make up for only a small portion of the screen [18].

This problem is solved when using IoU.

$$IoU = \frac{AreaofOverlap}{AreaofUnion} \tag{5}$$

In formula 5 you can see the formula used for computing the IoU. The $AreaofOverlap$ is equal to tp (true positive) since this is where the predicted labels and the source labels are the same. $AreaofUnion$ is equal to tp + fp (false positive) + fn (false negative). This is the union of all the predicted and source labels.

We can now rewrite the equation as:

$$IoU = \frac{tp}{tp + fp + fn} \tag{6}$$

Our goal is to determine the change in IoU by using CIConv compared to not using CIConv.

4 Experiments

We investigate to what degree CIConv improves a CNN's performance for semantic segmentation in the domain adaptation from day to night setting.

4.1 Experimental Setup

We have selected a subset of the Cityscapes [6] dataset for the source dataset. Since Cityscapes is a rather large dataset (3000 images), we instead opted for a smaller dataset, where we selected 200 images from the Cityscapes dataset, which we call: Minicity. Although using to fewer data led to suboptimal results, it was still crucial for us to operate this way, so the training time was feasible for us to work with. For the target dataset, we will be testing it on the Nighttime Driving dataset [7]. The chosen datasets represent a day-night domain adaptation scenario.

The experiment will be run in two different settings and with two experiments per setting.

Zero Shot Setting This will be used for our baseline. The [13] codebase will be used using the RefineNet architecture with ResNet-101 [12] feature extractors pre-trained on the ImageNet [8] dataset. We perform training with 175 epochs using SGD with momentum 0.9, weight decay 1e-4, and an initial learning rate of 0.1, which is step-wise reduced by a factor of 0.1 after every 30 epochs. All input images are resized to 1024x512 pixels and randomly cropped to 768x384 pixels, allowing a batch size of 3 on a NVIDIA GeForce GTX 1080 Ti GPU. Data augmentation is applied by random scaling, brightness-, contrast- and hue-shifting, and horizontal flipping. Inference is done on 1024x512 samples without cropping [13]. This setting is run once with a CIConv layer and once without a CIConv layer.

Unsupervised Domain Adaptation Setting This [19] codebase will be used using the DeepLab-v2 [5] framework with ResNet-101 [12] model pre-trained on ImageNet [8] as the segmentation baseline network. We perform training with 175 epochs using SGD with momentum 0.9, weight decay 5e-4, and a learning rate of 2.5e-4. All input images are resized to 1280x720 pixels, and all target images are resized to 1024x512 pixels, allowing for a batch size of 1 on a NVIDIA GeForce GTX 1080 Ti GPU. This batch size is also called stochastic mode: with this, the gradient and the neural network parameters are updated after each sample. We remove the last classification layer and modify the stride of the last two convolution layers from 2 to 1, making the resolution of the output feature maps effectively 1/8 times the input image size. To enlarge the receptive field, we apply dilated convolution layers in conv4 and conv5 layers with a stride of 2 and 4, respectively. After the last layer, we use the ASPP as the final classifier. Finally, we apply an up-sampling layer along with the softmax output to match the size of the input image [19]. This setting is run once with a CIConv layer and once without a CIConv layer.

4.2 Results

Results are shown in 1 as Intersection-over-Union. All the IoU's that have performed best in their specific label have been highlighted. With a mIoU of 0.303, regular AdaptSegNet significantly outperforms all the other trained models, including AdaptSegNet with a CIConv layer builtin.

Qualitative segmentation results are shown in figure 2. Most notable here is that (f) W-AdaptSegNet loses a lot of different features compared to (e) AdaptSegNet. Furthermore

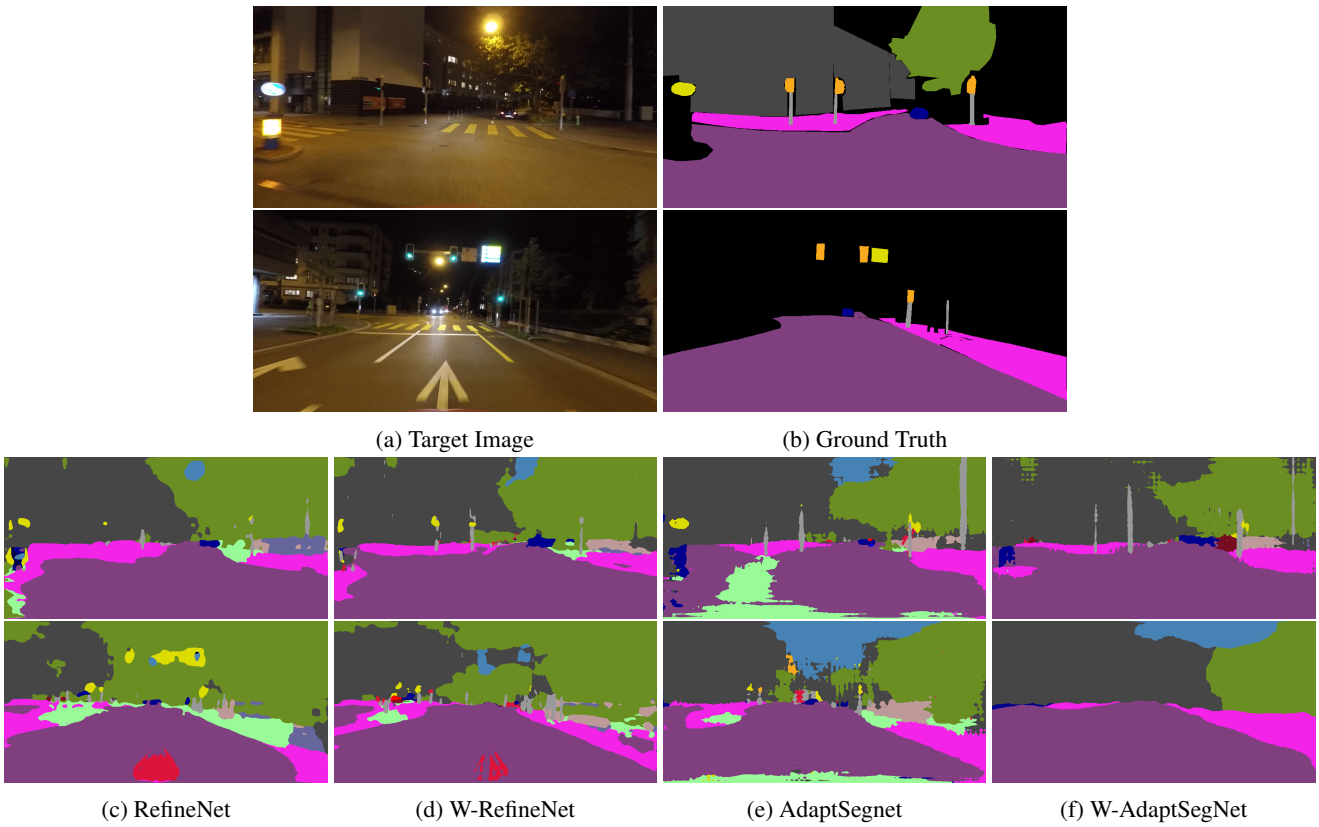
In figure 3 you can see the graph of the obtained mIoU for the different number of epochs. Notable here is that all of the methods do not substantially increase in performance after 40 epochs.

5 Discussion

As can be seen in table 1, from all tested methods, AdaptSegNet without a CIConv layer has performed the best from all tested methods. The IoU calculated for *sky* is remarkable for this because AdaptSegNet has performed astonishingly high in this label compared to the other methods. The sky in the daytime tends to have a blueish/whitish color which is a very different appearance than in nighttime, where the sky usually has a blackish color. AdaptSegNet probably performed better than RefineNet for *sky* because AdaptSegNet also has trained on the target dataset. Therefore AdaptSegNet was already exposed to the fact that the sky has a very different color. However, W-AdaptSegNet in contrast, has not performed better like regular AdaptSegNet. This is because the color invariant edge detector ignores the color property. Therefore the change in color does not provide any significant learning result for *sky*.

Minicity to Nighttime Driving																					
Method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
Trained on source data only																					
RefineNet	0.83	0.42	0.77	**0.10**	0.00	0.23	0.00	0.17	0.29	0.00	0.22	0.42	0.00	0.57	0.00	0.00	**0.05**	0.00	0.19	0.237
W-RefineNet	**0.87**	**0.52**	0.78	0.09	0.00	**0.36**	0.00	0.16	0.37	0.00	0.29	0.35	0.00	**0.63**	0.00	0.00	0.04	0.00	0.23	0.246
Trained on source and target data																					
AdaptSegNet	0.77	0.44	**0.81**	0.05	0.00	0.35	**0.14**	**0.47**	**0.53**	0.00	**0.73**	**0.46**	0.02	0.51	0.00	**0.06**	0.01	0.00	**0.40**	**0.303**
W-AdaptSegNet	0.81	0.35	0.70	0.06	0.00	0.29	0.04	0.35	0.35	0.00	0.26	0.40	**0.07**	0.46	0.00	0.00	0.00	0.00	0.21	0.228

Table 1: Results of adapting Minicity to the Nighttime Driving dataset expressed in IoU for every possible outcome label.



(a) Target Image (b) Ground Truth

(c) RefineNet (d) W-RefineNet (e) AdaptSegnet (f) W-AdaptSegNet

Figure 2: Example results of adapted segmentation for Minicity to Nighttime Driving. For each target image, we show RefineNet and AdaptSegNet with both with (d, f) and without (c, e) a CIConv layer.

Unlike [13], our results do not show any reliable or significant benefit from the CIConv layers. This may have multiple causes, such as using too small of a dataset for the model to learn with or having unoptimized hyper-parameters used to train. Having used a relatively small dataset might have led to a lack of generalization and difficulty in optimization, but it was necessary to use this smaller dataset due to time constraints.

Furthermore, most of the hyper-parameters used in our experiment have not differed from the original source code. An example of this is using random cropping and jitter in the zero shot setting, but not in the unsupervised domain adaptation setting. Another example is the difference in learning rate and weight decay between both of the settings. This may

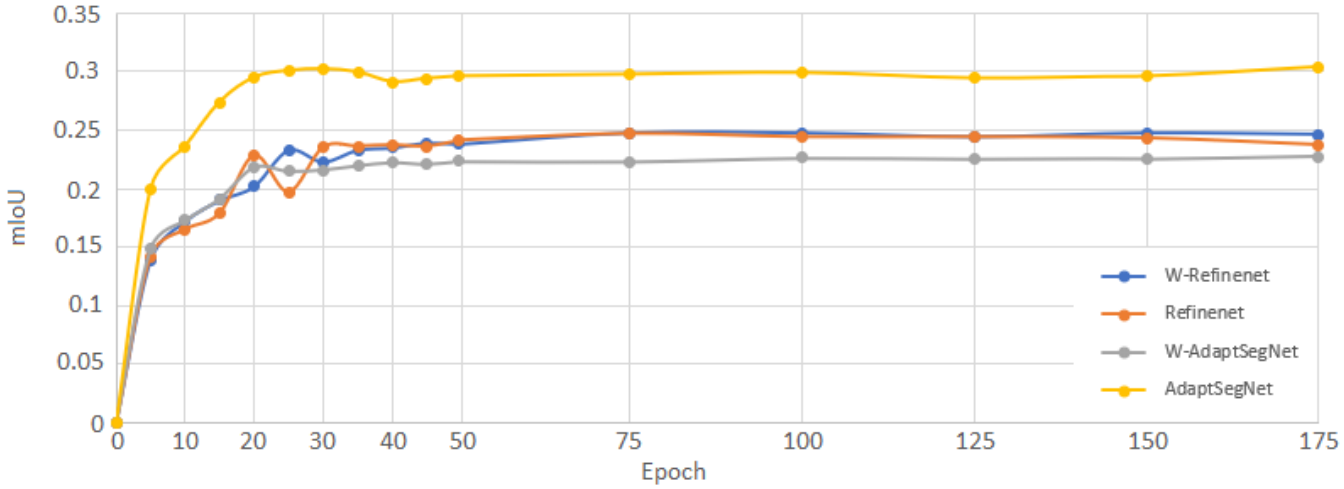


Figure 3: Calculated mIoU vs Epoch

have led to sub-optimal results, which have caused a lower mIoU, especially for experiments with CIConv. A handful of hyper-parameters have been changed from the original source code. These include settings like batch size and image size. These have been changed in order to properly run the code on the amount of memory that was made available.

More unlike [13], The mIoU for all experiments except for the vanilla AdaptSegNet have performed very similarly. In [13] W-Refinenet has performed better than AdaptSegNet, whereas in our results, the opposite is the case. Again, this is most likely due to adjusting the source dataset to a smaller one while keeping the same target dataset. This might have impacted the zero-shot approach more than the unsupervised domain adaptation approach since the unsupervised domain adaptation still has the same target data to train on.

6 Concluding Remarks and Future Work

This paper has analyzed the effectiveness of CIConv in an unsupervised domain adaptation setting and compared it to a zero-shot setting. In section 4 you can see the results of CIConv in these settings. CIConv has not led to a higher mIoU in the tested settings, and this may have several causes, such as having used a too small dataset or having used incorrect hyper-parameters during training.

Furthermore, CIConv is not immediately usable for unsupervised domain adaptation. This means that simply implementing a trainable CIConv layer into CNN is not sufficient for training might happen unoptimally. Therefore tweaking the hyper-parameters needs to be done before optimal conclusions can be derived from the tests.

As for future work, the same experiment could be executed but with different larger data sets. For example, Cityscapes can be used as source data instead of Minicity. The hyper-parameters can also be adjusted in future experiments as these likely resulted in sub-optimal results in our work. Other work that could be executed in the future is the implementation of CIConv in multiple different codebases, as we currently only have done in one single code base.

7 Responsible Research

This research does not involve any ethical aspects as no sensitive data is involved. However, everything that has been used is based on repositories and datasets from other authors. Therefore it is essential to credit all the previously done work. All the data used for this research is already publicly available and has been adequately referenced. All code used for this research from other repositories has also been credited and referenced.

If this research is going to be used in the real world (e.g., for self-driving cars), then ethical aspects have to be discussed. When utilizing the current code for the real world, further tweaking and testing will be needed. Since the model does not work 100% accurately, it is likely that vehicles will not execute optimally with the results, and therefore lives may be at risk.

7.1 Reproducibility

All the datasets used in this research will *not* be made available on our GitLab. This is because all the datasets are already publicly available for anyone to use. A simple search on the internet should give enough information for the user to download the used datasets. All the used code is available on GitLab. To ensure the reproducibility of the code, we recommend reading through section 4 as certain settings such as hyper-parameters have been specified in these sections. Since all code used is based on other repositories, we also recommend looking through these repositories if anything does not work as intended.

References

[1] M. Afifi and M. S. Brown. What else can fool deep learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 243–252.

[2] N. Alshammari, S. Akcay, and T. P. Breckon. On the impact of illumination-invariant image pre-transformation

for contemporary automotive semantic scene understanding. *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1027–1032, 2018.

[3] N. Alshammari, S. Akcay, and T. P. Breckon. Multi-task learning for automotive foggy scene understanding via domain adaptation to an illumination-invariant representation. *ArXiv, abs/1909.07697*, 2019.

[4] Gertjan J Burghouts and Jan-Mark Geusebroek. Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 2009.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR, abs/1606.00915*, 2016.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Dai D. and Gool L. V. Dark model adaptation: Semantic image segmentation from daytime to nighttime. *ITSC*, pages 3819–3824.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *Skolkovo Institute of Science and Technology (Skoltech)*, 2015.

[10] 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.

[11] Jan-Mark Geusebroek, Anuj Dev, Rein van den Boomhaard, Arnold W.M. Smeulders, Frans Cornelissen, and Hugo Geerts. Color invariant edge detection. *Scale Space Theories in Computer Vision, LNCS 1682, pp. 459-464*, 1999.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[13] Attila Lengyel, Sourav Garg, Michael Milford, and Jan C. van Gemert. Zero-shot day-night domain adaptation with a physics prior.

[14] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. 2016.

[15] Wulfmeier M. and Posner I. Bewley. Addressing appearance change in outdoor robotics with adversarial domain adaptation. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1551–1558.

[16] P.Kubelka and F. Munk. Ein beitrag zur optik der farbanstrichen. *Zeitung fur Technische Physik volume 12*, page 593, 1999.

[17] B. Sahu. The evolution of deeplab for semantic segmentation. https://towardsdatascience.com/the-evolution-of-deeplab-for-semantic-segmentation-95082b025571, 2019.

[18] E Tiu. Metrics to evaluate your semantic segmentation model. https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2, 2019.

[19] YiHsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. 2020.

[20] Ben Upcroft, Colin McManus, Winston Churchill, William P. Maddern, and Paul Newman. Lighting in invariant urban street classification. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1712–1718, 2014.

[21] Çağrı Kaymak and Ayşegül Uçar. A brief survey and an application of semantic image segmentation for autonomous driving. *Neurocomputing, 2018, 312: 135-153*, 2018.