On the impact of rain over semantic segmentation of street scenes

Simone Zini[1 [0000−0002−8505−1581]] ✉ and Marco Buzzelli[1 [0000−0003−1138−3345]]

Department of Informatics, Systems and Communication
University of Milano – Bicocca
s.zini1@campus.unimib.it, marco.buzzelli@unimib.it

Abstract. We investigate the negative effects of rain streaks over the performance of a neural network for real time semantic segmentation of street scenes. This is done by synthetically augmenting the CityScapes dataset with artificial rain. We then define and train a generative adversarial network for rain removal, and quantify the benefits of its application as a pre-processing step to both rainy and "clean" images. Finally, we show that by retraining the semantic segmentation network on images processed for rain removal, it is possible to gain even more accuracy, with a model that produces stable results in all analyzed atmospheric conditions. For our experiments, we present a per-class analysis in order to provide deeper insights over the impact of rain on semantic segmentation.

Keywords: Rain removal · data augmentation · semantic segmentation.

1 Introduction

The automotive field has seen a strong expansion in recent years, where a crucial role is played by perception systems for autonomous vehicles and for assisted driving. The development of computer vision techniques in this field potentially allows for a decrease of the production costs due to the exploitation of inexpensive hardware, i.e. RGB cameras in place of depth sensors. Large benchmark datasets such as the CityScapes dataset have proven to be extremely valuable in developing and testing automotive-related solutions, such as networks for monocular depth estimation [1] and for semantic segmentation [10] of street scenes.

The specialized literature is mostly focused on either improving the model accuracy, or in reducing the computational complexity of the involved models. Relatively little effort has been put into investigating and quantifying the impact of meteorological conditions over the method performance, including phenomena that alter the image quality such as haze, rain, and changes in illumination conditions. This is mostly due to the lack of appropriate datasets, i.e. real-life photos acquired in bad weather conditions, and annotated for computer vision tasks such as semantic segmentation. For this reason, in fact, we will resort to synthetic rain augmentation over annotated datasets for the quantitative experiments presented in this paper.

Valada et al. [17] presented a multi-stream deep neural network that learns features from multimodal data, and adaptively weights different features based on the scene conditions. The authors assessed semantic segmentation on the synthetic dataset Synthia [15] (which includes rainy scenes), but did not offer a direct comparative evaluation on the presence and absence of rain-induced artifacts. Khan et al. [6] created an entirely artificial dataset for semantic segmentation in different atmospheric conditions, to be used as training data for the task. We argue that although synthetic data generation is essential in producing an adequately large database, introducing synthetic rain artifacts over real images would instead offer the grounds for an evaluation that is closer to a real-case scenario. Porav et al. [13] focused on the removal of rain droplets, which by definition refer to the artifacts introduced by a wet glass on a clear day. As such these are only partially representative of real-case scenarios. Halder et al. [3] developed a physics-based data augmentation technique, used to train more robust models for semantic segmentation, although they offer no insights on the benefits of rain-removal techniques. Recently, Li et al. [7] defined a unified benchmark for images perturbed by rain streaks, rain drops, and mist, and tested different methods for rain removal, including among the evaluation criteria the impact over vehicle detection. In terms of general purpose rain removal, without emphasis on its impact over other computer vision tasks, the scientific literature offers a wide variety of solutions. A recent review by Yang et al. [18] presents a comprehensive analysis of methods, from model-based to data-driven.

In this paper, we focus on the effects of rain streaks on a neural network for semantic segmentation of street scenes. We experiment with a semantic segmentation architecture that belongs to the class of "real time" solutions, as we consider it to be an interesting use case for a stress test, whereas accuracy-oriented networks would be implicitly more robust. The domain of semantic segmentation inherently enables a fine-grained analysis of the results, as per-class performance can be individually scrutinized. The aforementioned lack of annotated datasets for semantic segmentation in a rainy environment is here addressed by performing synthetic data augmentation, in the form of artificial rain streaks. Additionally, we are interested in evaluating to what extent it is possible to regain semantic segmentation accuracy by exploiting rain removal techniques. For this reason, we introduce a generative model for the removal of rain streaks, to be applied over the synthetically-augmented images.

2 Methodology for rain generation, rain removal, and semantic segmentation

In this section we describe the building blocks of our study over the impact of rain on semantic segmentation of street scenes. We first describe how artificial rain streaks can be generated and introduced on existing datasets. We then define a generative adversarial network for the removal of rain in digital images, and finally we define the adopted encoder-decoder network for real-time semantic segmentation.

In this and the next sections, we will call the images clean, rainy, and rain-removed in relation to their status: with clean images, we will refer to images taken in clear weather conditions, with rainy images, we will refer to pictures taken under rainy weather, and finally, with rain-removed images, we will refer to images processed in order to remove the rain originally presents in the picture.

2.1 Synthetic Rain Augmentation

In order to reproduce semi-realistic rainy images, following the work done in [20], we decided to generate random rainy masks to apply over the target images. Differently from [19], instead of using Photoshop to generate a limited number of masks to randomly apply to the images, we used MATLAB to create a random rainy mask generator, which for each image generates a new mask, based on some parameters randomly selected in ranges that have been defined empirically, with respect to the original approach from [19] and the objective of obtaining semi-realistic rainy images. The pipeline is represented in Figure 1.

Starting from an sRGB image, the process first generates a raindrop mask, by choosing four parameters: d_1 rain density, σ_1 Gaussian filter dimension, l_1 streak length, α_1 falling angle. After that, a rain streaks map is generated using two parameters previously chosen for the first mask: l_1 streak length and σ_1 falling angle, and two other ones chosen at this step: d_2 rain density and α_2 Gaussian filter dimension. Eventually, an optional haze mask is generated. These three masks are then applied to the image in order to obtain the rainy version of the original input image.

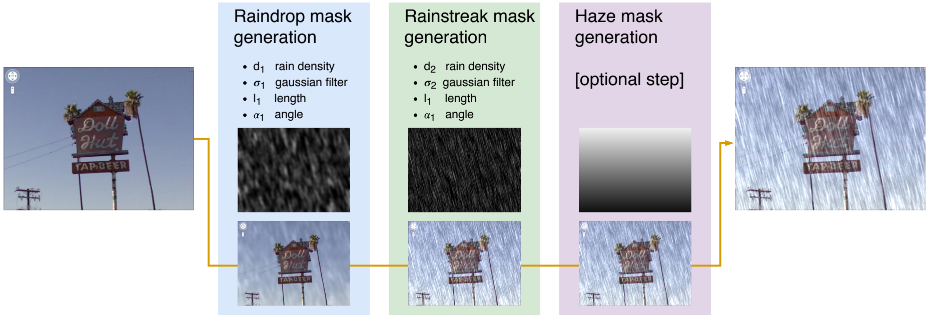


Fig. 1: Steps of the pipeline designed for the synthetic rain generation

2.2 Generative Adversarial Network for Rain Removal

In order to perform the rain removal process we employed an autoencoder Convolutional Neural Network (CNN) based on the U-Net architecture, trained in a conditional Generative Adversarial Network (GAN) framework. In particular we adopted the method used for rain reduction in [20].

The structure of the rain removal CNN is based on the U-Net [14] architecture, with the addition of skip connections as done for Pix2Pix network [4]. The architecture is shown in Figure 2. Based also on recent works related to image enhancement, some changes have been made to the classical U-Net architecture, in order to reduce the introduction of artifacts and improve the quality of the final results:

- The normalization layers have been removed from the model, in order to avoid the generation of artifacts, as done in [8] and [12].
- Max-pooling operation have been replaced with convolutions with 2-pixel stride to reduce feature spatial dimensions, without losing useful information for the restoration process.
- A combination of bilinear upsampling with 2D Convolution has been adopted, to reduce artifacts coming from the application of the deconvolutional layers in the decoder part of the network.

In order to train the model in a Generative Adversarial Network framework we adopted a patchGAN discriminative network, trained in a Conditional GAN training approach [11]. The architecture of the discriminative model is shown in Figure 2. During the training phase the discriminative network is fed with the concatenation of the enhanced image and the original input image. The overview of the network combination for training is shown in Figure 3.

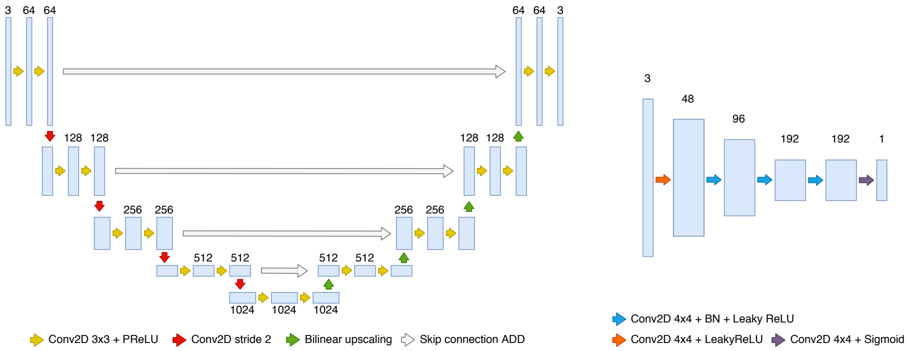


Fig. 2: Generative and discriminative model architectures. The generative model is a U-Net autoencoder style architecture: the max pooling layers have been replaced with convolutions with strides>1 and the upscaling operation is performed with Bilinear Interpolation combined with convolutions. The discriminative network is based on the architecture of the Conditional PatchGAN discriminator.

Loss Function The loss function used to train the model is defined as:

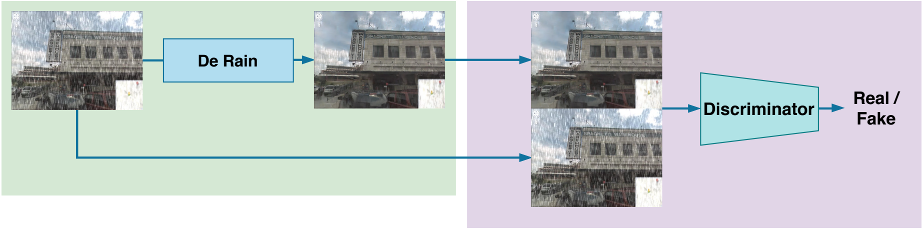


Fig. 3: Training system with Conditional patchGAN.

$$Loss = \lambda_e \cdot L_e + \lambda_{adv} \cdot L_{adv} + \lambda_p \cdot L_p. \tag{1}$$

which is the combination of three loss functions, weighted by three different weight values $\lambda_e, \lambda_{adv}, \lambda_p$

Given an image pair $\{x, y\}$ with C channels, width W and height H (i.e. $C \times W \times H$), where x is the input image and y is the corresponding target, we define the three loss function as follows.

The per-pixel Euclidean loss, defined as:

$$L_e = \frac{1}{CWH} \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} ||\phi_E(x^{c,w,h}) - y^{c,w,h}||_2^2, \tag{2}$$

where $\phi_E(\cdot)$ is the learned network for rain removal.

The perceptual loss [5] defined as distance function between features extracted from the target and output images, using a pre-trained VGG network [16]:

$$L_p = \frac{1}{C_i W_i H_i} \sum_{c=1}^{C_i} \sum_{w=1}^{W_i} \sum_{h=1}^{H_i} ||V(\phi_E(x^{c,w,h})) - V(y_B^{c,w,h})||_2^2, \tag{3}$$

where $V(\cdot)$ represents a non-linear CNN transformation (VGG16 network).

Finally, the original GAN loss described as:

$$L_{adv} = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_x[\log(1 - D(x, G(x)))], \tag{4}$$

where $G(\cdot)$ is the trained generative network for image de-raining.

2.3 Encoder-decoder Network for Semantic segmentation

Semantic segmentation integrates traditional image segmentation (which partitions the input image into subregions without a regular shape) with the categorical classification of each generated subregion. The great majority of current image segmentation architectures are based on the encoder-decoder structure [9] to exploit a large receptive field and, at the same time, to produce high-resolution predictions.

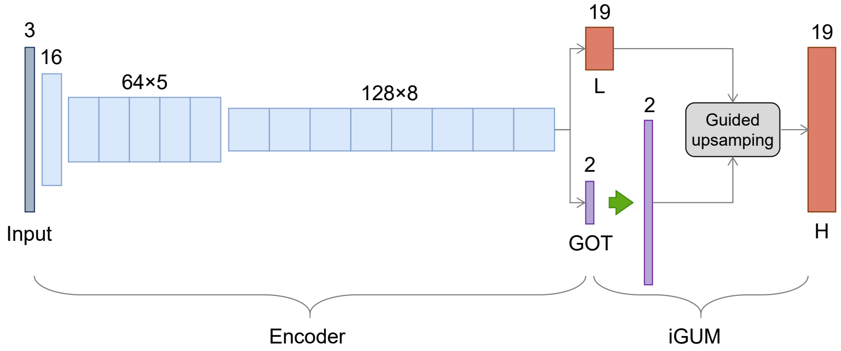


Fig. 4: Schematic representation of the iGUM encoder-decoder architecture used for real-time semantic segmentation. Each activation is annotated with the corresponding number of channels.

We focus our study on the iGUM network (improved Guided Upsampling Module) [10]. The architecture, shown in Figure 4, is composed of a lightweight encoder and a iGUM-based decoder.

The encoder structure is compact and thus efficient. Its main building block is a lightweight non-bottleneck residual block with two asymmetric kernels (3×1) and (1×3) preceded and followed by 1×1 channel-wise convolutions. The encoder produces two low-resolution outputs: the probability map per class (L), and a low-resolution Guidance Offsets Table (GOT), which is then upsampled to the target resolution by means of a scaling factor f. Both branches serve as input to the iGUM module.

In encoder-decoder architectures, the decoder typically plays a refinement role where features are iteratively upsampled to match the input size and to finally output a dense per-pixel prediction. In this case, the decoder is effectively replaced *in toto* by the iGUM module: whereas semantic segmentation traditionally involves the prediction of a dense per-pixel map of class probabilities, the iGUM module introduces a more efficient representation, which allows for a non-uniform density grid. Let $L \in \mathbb{R}^{N\times M\times C}$ be the low-resolution C-channel probability map, and let p_i and q_i be the two channel composing the Guidance Offsets Table at high resolution. The output of iGUM, a high-resolution map $H \in \mathbb{R}^{fN\times fM\times C}$, is computed as a generalization of the nearest neighbor operator:

$$H_i = \sum_n^N \sum_m^M L(\delta(\lfloor x_i^s + p_i + 0.5\rfloor - m), \delta(\lfloor y_i^s + q_i + 0.5\rfloor - n)) \tag{5}$$

Indices n and m slide over the low-resolution dimensions, while index i is used to indicate the high-resolution domain. x_i^s and y_i^s are the spatial sampling coordinates. $\lfloor x_i^s + 0.5\rfloor$ and $\lfloor y_i^s + 0.5\rfloor$ round the coordinates to the nearest integer location, and δ is a Kronecker function for input selection. Offsets p_i and q_i

effectively shift the sampling coordinates of each grid element in the x and y dimensions respectively.

3 Experiments

3.1 Dataset and evaluation metrics

We performed our experiments on the Cityscapes [2] dataset: a set of urban street images annotated with pixel-wise semantic information. It is composed of 5000 high-resolution images (2048×1024) out of which 2975, 500 and 1525 images belong respectively to train, validation and test subsets. Annotations include 30 different classes of objects, although only 19 are typically used for training and evaluation, plus a background class:

- road
- sidewalk
- building
- wall
- fence
- pole
- traffic light
- traffic sign
- vegetation
- terrain
- sky
- person
- rider
- car
- truck
- bus
- train
- motorcycle
- bicycle
- (background)

The dataset is characterized by a vast diversity of scenes, with images taken from different cities all with good or medium weather conditions.

Two metrics are used for model validation: avereage of class-wise Intersection over Union (IoU, also called Jaccard Index) and average class-wise Accuracy. These are computed as:

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{6}$$

$$\text{Accuracy} = \frac{TP}{TP + FN} \tag{7}$$

Where TP, FP and FN are, respectively, the number of True Positive, False Positive, and False Negative pixels.

3.2 Experimental Results

In this section we are going to analyze the performance of the semantic segmentation model in relation to the condition of the data involved (i.e. "clean", rainy, and rain-removed). We evaluate the model considering condition of data used for training as well as for validation.

For the analysis we have considered three version of the Cityscapes dataset:

- **Clean images**: the original images from the Cityscapes dataset.
- **Rainy images**: images obtained using the synthetic mask generation algorithm starting from the "clean" images.
- **Rain-removed images**: images obtained by removing the rain from the rainy images version of the Cityscapes, using the rain reduction algorithm.

This analysis has been done with the purpose of studying how the level of degradation in data can affect the performances of the segmentation algorithm in inference time, and how it can affect the learning process of the model.

Table 1 and 2 report respectively the mean Accuracy and mean of class-wise Intersection over Union.

Table 1: Accuracy of semantic segmentation on the Cityscapes validation dataset: table shows the results in relation to the training data used for the semantic segmentation network.

Accuracy		Test data		
		Clean	Rainy	Rain removed
Training data	Clean	72.88%	24.73%	41.57%
	Rainy	35.34%	35.75%	34.66%
	Rain removed	69.75%	67.00%	67.96%

Table 2: Intersection Over Union of semantic segmentation on the Cityscapes validation dataset: table shows the results in relation to the training data used for the semantic segmentation network.

IoU		Test data		
		Clean	Rainy	Rain removed
Training data	Clean	62.59%	15.00%	27.50%
	Rainy	29.48%	29.31%	27.85%
	Rain removed	58.03%	56.28%	57.64%

As can be seen from the tables, for what concerns the segmentation with the model trained on “clean” (rain-free) images, the rain removal step helps to improve the performance with respect to direct segmentation of rainy images. While, as expected, with the “clean” validation images the segmentation algorithm performs better than the other cases. It is interesting to notice how the rain-removal pre-processing operation brings to an accuracy improvement of 16.84% and mIoU of 12.50% between the rainy images and the rain-removed ones.

For what concerns the other two training cases, the one with the rainy training set and the one with the rain-removed training set, we can observe a different behavior. In both of the cases, independently from the validation dataset used, the performance of the segmentation model in terms of accuracy an IoU does not change significantly. This behavior can be related to the amount of information present in the images used for training. In the first of these two cases, the network trained with the rainy images is not able to perform better than 36% in terms of accuracy, and 30% in terms of mIoU. Since during the training phase, part of the information in each image is always occluded or corrupted, the model is not capable to learn correct feature extraction for the classification of some specific classes. Looking at the per-class mIoU analysis in Figure 5b, some of the classes have mIoU of 0% or values very near to zero. As can be seen, for the three validation set the situation is the same for all the classes, behavior that shows how the limited capability of the network to correctly segment element of the images is related to the missing information during training time, and not related to the type of validation data.

Similar behavior can be observed with the model trained with the rain-removed images. In this case, the performance improves in terms of accuracy and IoU due to the partially restored information in the training set, after the use of the rain-removal model. However, the general behavior is the same as the previous case: even if we test the model with "clean" images, the model is not able to perform better than the other two validation set cases, due to the missing knowledge in the training images.

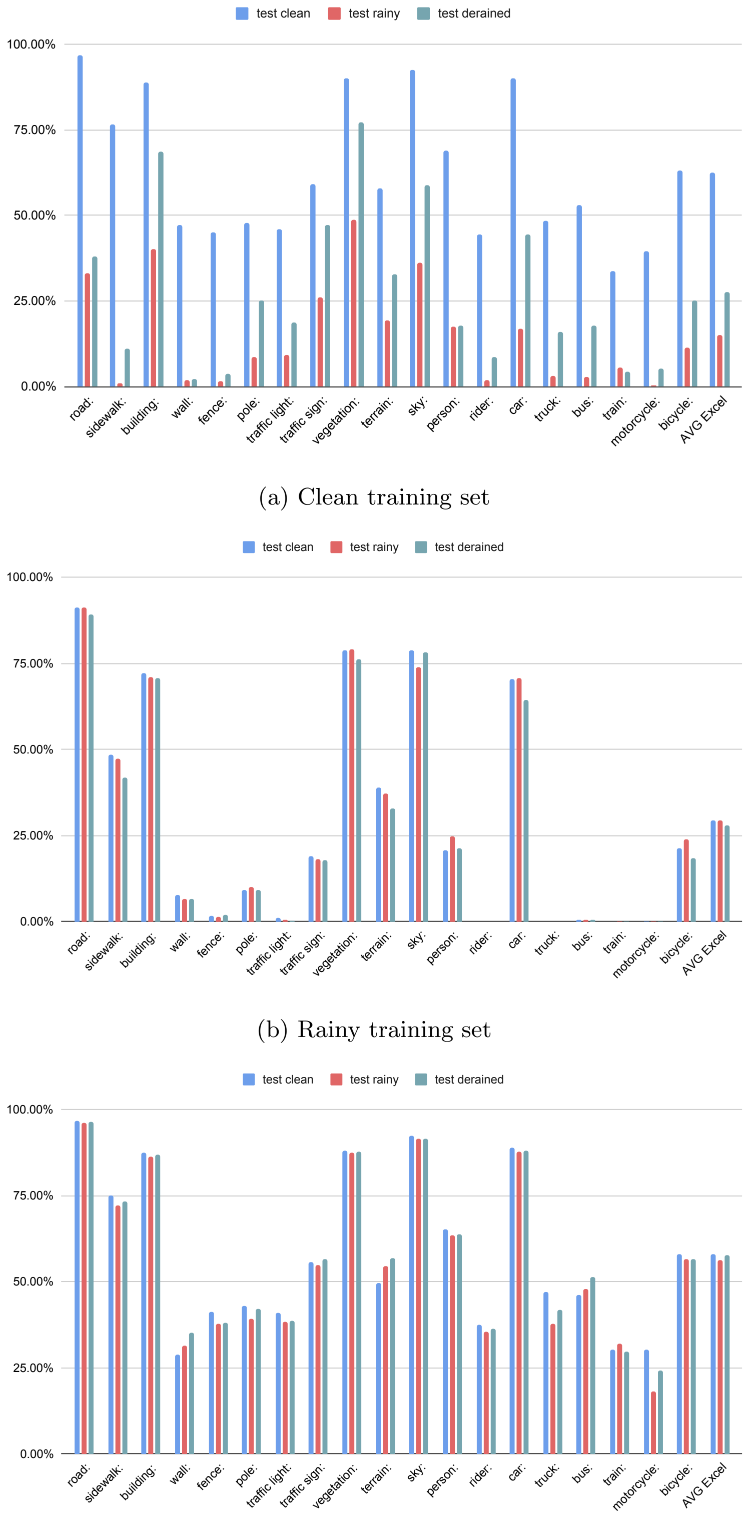
In Figure 5c we can see the per-class analysis: it is easy to observe the same behavior of the model trained with the rainy images, but it is also possible to observe an improvement in the segmentation of classes that were not recognized by the model trained with rainy images. This improvement is related to the enhancement of the training data due to the pre-processing step over the training set. Aside, it is interesting how the training with the rain-removed images has improved the results of the segmentation model with respect to the one trained with "clean" images.

3.3 Visual inspection

We here present a visual inspection of the impact of rain and rain-removal techniques over semantic segmentation. Figure 6 clearly shows the deterioration in prediction accuracy introduced by rain-related artifacts (row c). The strong texture of rain streaks completely changes the interpretation of the road and sidewalk areas, which are mistaken as an obstacle (wall/fence). This phenomenon occurs despite the strong intrinsic bias of the "road" class, that appears in the plurality of training data pixels, and that occupies a consistent area throughout different images. On top of this, small regions such as the traffic signs and far-away vehicles are completely missed.

By processing the rain-augmented image using our rain-removal network, it is possible to partially restore the accuracy of semantic segmentation in some of the areas. As Figure 6d shows, the segmentation of the small cars is almost completely recovered, and part of the road and sidewalk are correctly identified. A qualitative and subjective evaluation of rain removal on the RGB image shows arguably less impressive results, suggesting a disconnect between perceived quality and usefulness for computer vision.

The best results, however, are obtained by retraining the semantic segmentation network using images that were processed with the rain-augmentation pipeline and subsequent rain-removal. In this case, the prediction in the same



(a) Clean training set

(b) Rainy training set

(c) Rain-removed training set

Fig. 5: Mean intersection over union value for each class of the Cityscapes test set

scenario, as depicted in Figure 6e shows an excellent restoration of several details, although some imperfections remain in the top-right corner of the example image.

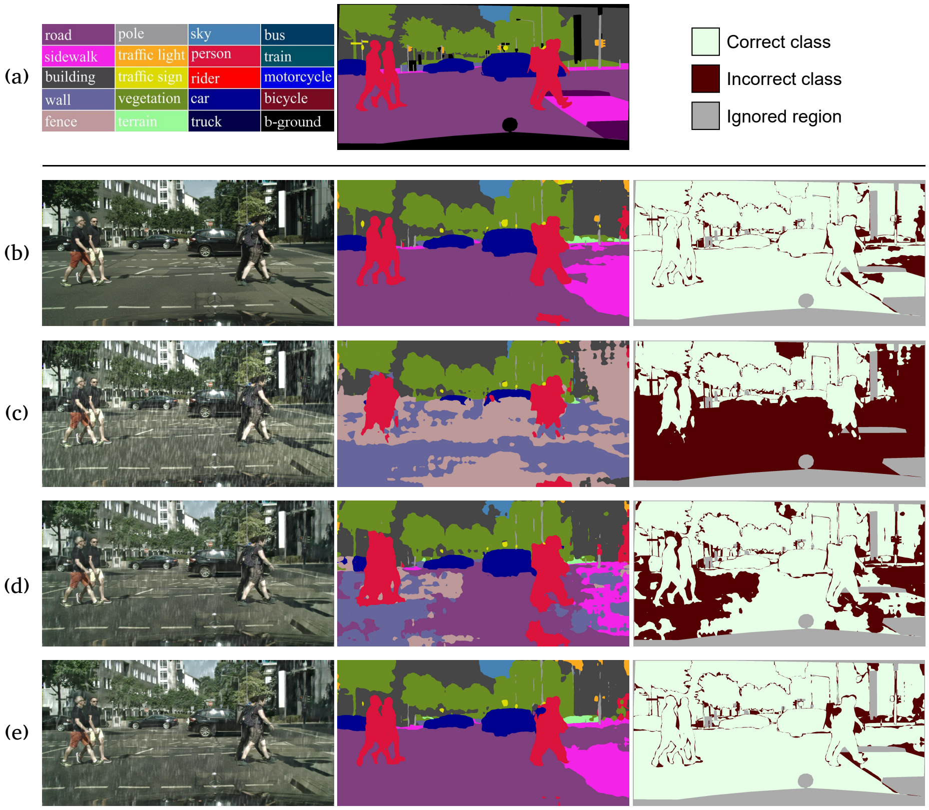


Fig. 6: Impact of rain on semantic segmentation. Row (a) presents the color coding for semantic segmentation, the ground truth for the analyzed image, and the legend for error visualization. Rows (b) to (e) show, respectively, the prediction on the original "clean" image, on the image with artificial rain, on the image with rain removed, and once again on the image with rain removed but using a semantic segmentation model trained on images processed for rain and subsequent rain removal.

For the sake of completeness, we also perform a qualitative evaluation over two out-of-dataset real-life pictures, depicted in Figure 7. Specifically, we report semantic segmentation results over the original rainy images trained with the "clean" version of the Cityscapes dataset (column a), and results over rain-removed images trained with the rain-removed version of Cityscapes (column b). These two extreme cases show the significant improvement in segmentation quality that can be obtained by the joint application of our rain removal network both on training data and inference data. The final results still show some imperfections, which can be attributed to the different nature of the image data when compared to the training set, both concerning rain appearance, as well as the general content and format of the pictures.

4 Conclusions

In this work we investigated the negative effects of rain streaks over the performance of neural networks. Specifically we focused on the semantic segmentation task.

We proposed an analysis of the performance of a semantic segmentation neural network, in relation to different training configurations, and different inference conditions.

In order to perform the analysis we defined a pipeline for synthetic rain generation and a rain removal neural network to augment the Cityscapes dataset, obtaining three different versions of the dataset for both training and testing the semantic segmentation network.

The model has been trained in three different conditions: with "clean" images, with images with artificial rain streaks, and with images processed for removal of the artificial rain streaks. In the first case the experiments show how the application of rain-removal on rainy images gives benefit for the segmentation step of a model trained in optimal image conditions. The other experiments, regarding the impact of the degraded information in the images used for training the model, shows how the application of an enhancement algorithm can improve the performance of the model at inference time. We observed an improvement of 34% of accuracy and 30% of mIoU between the model trained with the degraded images and the one trained with the enhanced ones.

References

1. Bianco, S., Buzzelli, M., Schettini, R.: A unifying representation for pixel-precise distance estimation. Multimedia Tools and Applications **78**(10), 13767–13786 (2019)
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223 (2016)
3. Halder, S.S., Lalonde, J.F., Charette, R.d.: Physics-based rendering for improving robustness to rain. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 10203–10212 (2019)
4. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
5. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)

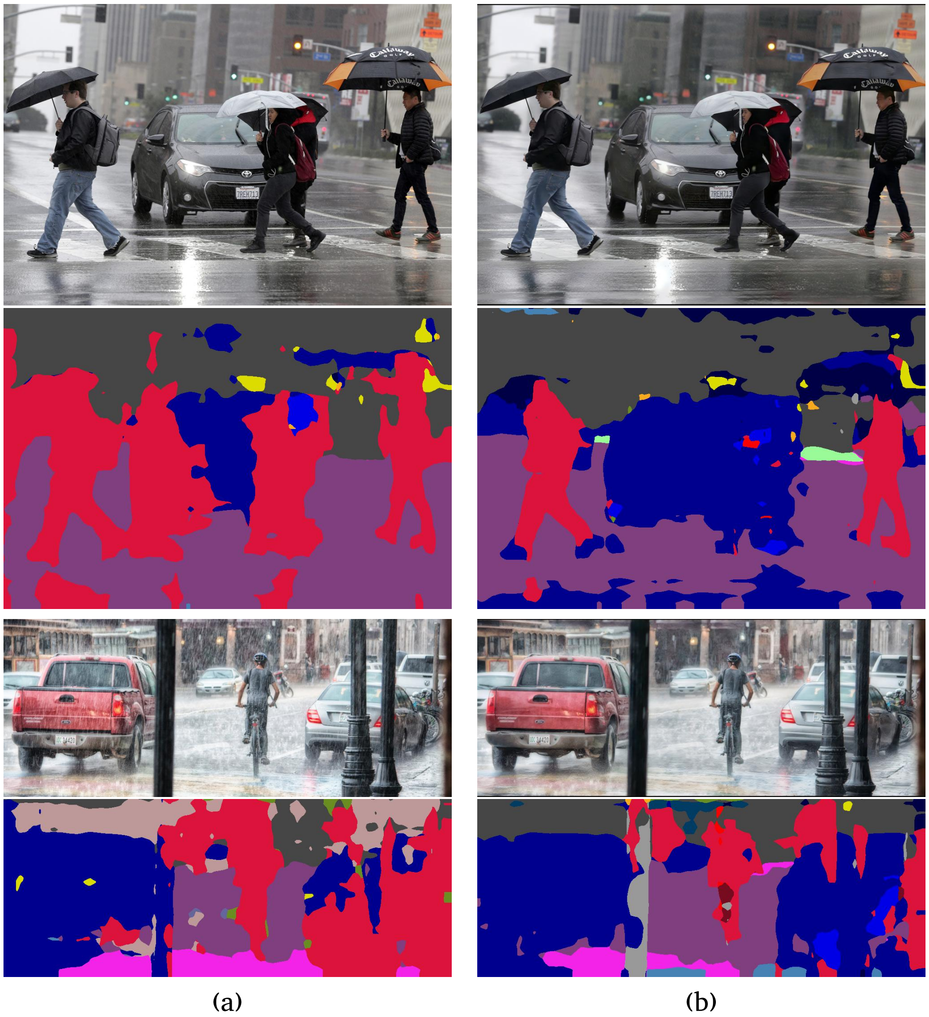


Fig. 7: Visual assessment of rain (column a) and rain-removal (column b) over real case images, using semantic segmentation trained respectively on “clean” images, and images processed for rain-removal. Original images credit Nick Út, and Genaro Servín.

6. Khan, S., Phan, B., Salay, R., Czarnecki, K.: Procsy: Procedural synthetic dataset generation towards influence factor studies of semantic segmentation networks. In: CVPR Workshops. pp. 88–96 (2019)
7. Li, S., Araujo, I.B., Ren, W., Wang, Z., Tokuda, E.K., Junior, R.H., Cesar-Junior, R., Zhang, J., Guo, X., Cao, X.: Single image deraining: A comprehensive benchmark analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3838–3847 (2019)
8. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 136–144 (2017)
9. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
10. Mazzini, D., Schettini, R.: Spatial sampling network for fast scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
11. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
12. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3883–3891 (2017)
13. Porav, H., Bruls, T., Newman, P.: I can see clearly now: Image restoration via deraining. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 7087–7093. IEEE (2019)
14. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
15. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3234–3243 (2016)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint **arXiv:1409.1556** (2014)
17. Valada, A., Vertens, J., Dhall, A., Burgard, W.: Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 4644–4651. IEEE (2017)
18. Yang, W., Tan, R.T., Wang, S., Fang, Y., Liu, J.: Single image deraining: From model-based to data-driven and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
19. Zhang, H., Sindagi, V., Patel, V.M.: Image de-raining using a conditional generative adversarial network. IEEE transactions on circuits and systems for video technology (2019)
20. Zini, S., Bianco, S., Schettini, R.: Cnn-based rain reduction in street view images. In: Proceedings of the 2020 London Imaging Meeting. pp. 78–81 (2020). https://doi.org/doi.org/10.2352/issn.2694-118X.2020.LIM-12