Shadow Removal with Paired and Unpaired Learning –Supplementary Material–

Florin-Alexandru Vasluianu
ETH Zürich
fvasluianu@student.ethz.ch

Andrés Romero
ETH Zürich
roandres@ethz.ch

Luc Van Gool
ETH Zürich
vangool@vision.ee.ethz.ch

Radu Timofte
ETH Zürich
radu.timofte@vision.ee.ethz.ch

Contents

In this supplementary material we provide more details about the methods presented in the main 'Shadow Removal with Paired and Unpaired Learning', along with supplementary quantitative and qualitative results, as follows:

i) Section 1 provides more **details of the proposed method**, and fully describes and differentiates the proposed schemes for unpaired and paired training settings.

ii) Section 2 provides **a more comprehensive ablation study**. Particularly, we report results with different configurations of losses, the influence of the number of training epochs, analysis of the shadow adder generator and its performance, and finally, the effect of different artifacts suppression strategies using data augmentation.

iii) Section 3 provides **supplementary visual results** on ISTD+ [2] and USR test images [1].

1. Proposed methods - more details

1.1. Architecture

Details of the *generator* implementation are provided in Table 1, where the operation o_1 is a convolutional operation with kernel size 4, stride 2 and padding 1 and o_2 is its upsampling counterpart that uses a transposed convolution. Skip-connections were added between the downsampling blocks and the upsampling counterparts. The o_3 operation, present in the last layer of the architecture, is a convolution with kernel size 4 and padding 1, preceded by an upsampling with scale factor 2 and zero padding 1 in the top and the bottom size of the feature tensor. The result is then passed into the $\tanh$ activation, obtaining the corresponding pixel in the produced image.

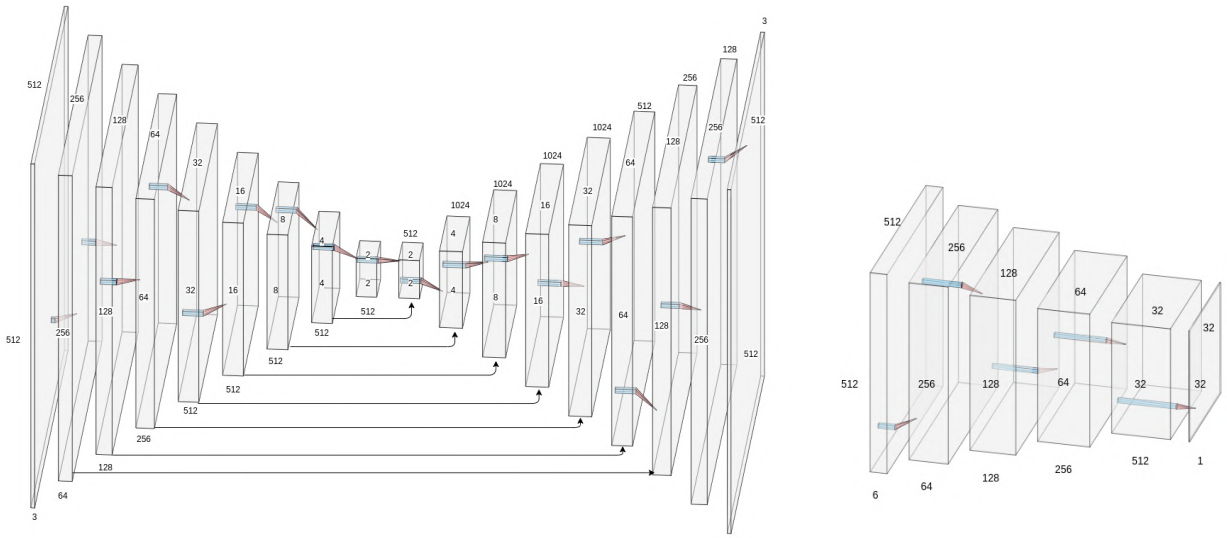
The *Shadow adder* and *Shadow remover* generator follow roughly the same architecture (see Figure 1), based on a downsampling part and its counterpart upsampling part where skip connections were added between the blocks in the downsampling half (encoder) to the corresponding blocks in the upsampling half (decoder). The only difference between those two components is the number of input channels. This is explained by the necessity of the shadow adder generator to have the shadow mask, as the localization information of the shadow-affected areas in the input image.

As the model is able to learn from unpaired data, and the true mask will not be provided along the training procedure, is very important that the masks provided as inputs (even if they are randomly sampled) to be realistic, and so, to be able to achieve enough control, the weights for the GAN loss and the content loss are to be increased. Also, by randomly sampling the negative examples for the training process of the discriminators, the generators will benefit from their better generalization ability, producing better results.

In Figure 3, we follow the transformations suffered by the input data along the cycles depicted in Figure 2, providing, for randomly sampled training examples, the outputs of the generators and the shadow masks used in order to compute each shadow affected image. As the method used to do the binarization was to perform a thresholding operation by the median value of the grayscale mapped image difference, the difference to the input shadow mask is consistent. However, this method induced the ability of the model to detect the steep variation in terms of pixel illumination, successfully detecting the real shadow-affected regions in the input image, avoiding producing a all-zero shadow mask, by simply applying an identity mapping.

Table 1: General details about the architecture of the generators. *LR* is LeakyReLU(0.2), *R* is ReLU, and *TH* is the hyperbolic tangent activation function.

Layer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Channels in	3/4	64	128	256	512	512	512	512	512	1024	1024	1024	1024	512	256	64
Channels out	64	128	256	512	512	512	512	512	512	512	512	512	256	128	64	3
Operation	o_1	o_1	o_1	o_1	o_1	o_1	o_1	o_1	o_2	o_2	o_2	o_2	o_2	o_2	o_2	o_3
Normalization	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
Activation	LR	LR	LR	LR	LR	LR	LR	LR	R	R	R	R	R	R	R	TH
Dropout	0	0	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0



a) generator b) discriminator

Figure 1: The architecture of our generators and discriminators. For shadow adder generator G_s, the initial number of channels will be 4 (the mask is concatenated to input image).

1.2. Overall scheme description

As shown in Figure 2, the architecture of the proposed solution presents five major components. The first four are paired in such a way that, for the image pair (u, v), with X the shadow images domain, and Y the shadow free images domain, $u \in Y$ and $v \in X$, the *shadow adder generator* will learn the mapping $G_s(u, m)$ and the *shadow discriminator* D_s will distinguish between pairs of images in the X domain and pairs with at least one image in the Y domain.

Similarly, the *shadow remover generator* will learn the mapping $G_f(v)$ and the *discriminator free* will ask the generator for better qualitative results in terms of the properties characterizing the shadow removal mapping.

The *perceptual loss module*, based on the VGG-16 architecture, will use high-level features to help the generators to produce better qualitative results, in terms of color loss, style loss, and content loss.

So, for the (u, v) image pair, along with the mask $m = Bin(u - v)$, the intermediate representation $(\hat{u}, \hat{v})$ will be computed such that $\hat{u} = G_f(v)$ and $\hat{v} = G_s(u, m)$. The m variable is the shadow mask, a 1-channel image where the value $m_{i,j} = 1$ means that in the u image the pixel $u_{i,j}$ is a shadow affected pixel. So, the shadow mask provides the shadow localization information, and, by providing both the localization information and the target shadow free image, the problem can be solved like a regression task. This was referenced as the *forward step* of the cycle.

The results of this step will be used in the image recovering procedure, a corresponding reconstruction step in both cycles implemented, such that having the synthetic shadow mask $\hat{m^f} = Bin(\hat{u} - v)$, the images u_r, v_r can be produced by applying the transformations G_f and G_s, respectively $u_r = G_f(\hat{v})$ and $v_r = G_s(\hat{u}, \hat{m^f})$. The cycle consistency is enforced on the computed recovered results, in both domains X and Y, as $u_r = u$ and $v_r = v$.

The shadow mask is computed as a difference between

a shadow free image and a shadow affected image, as the shadow image is characterized by the property that in the shadow image the intensity of the pixels is lower than in the shadow free counterpart. The result is then transformed in the *grayscale* image representation and the Bin function is applied, as the binarization using as threshold the median value of the grayscale image difference.

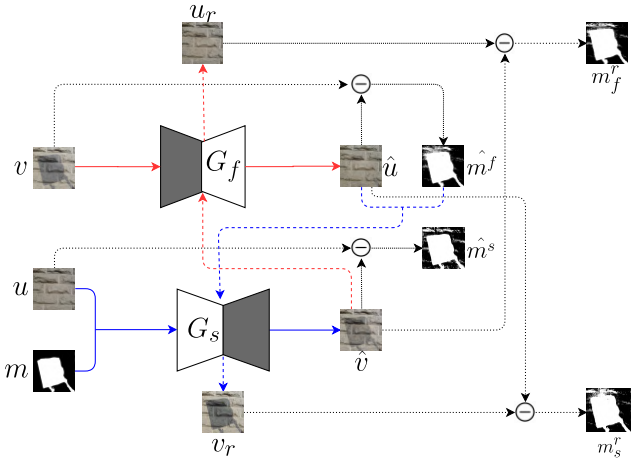


Figure 2: Overall scheme of the proposed solution. As a convention, red lines were used for the shadow removal step, blue lines for the shadow addition. As contiguous lines were used for the forward step, dashed lines represent the reconstruction step. Black dotted lines were used for the binary mask computation.

The components in the training objective will be computed using the results represented in the Figure 2, as adversarial, perceptual, mask and pixel-wise penalties applied on both the results of the forward step and the recovered images. The reasons for choosing them will be listed as follows. The generated image pairs $(v, \hat{v}), (v, v_r)$ should be drawn from the $X \times X$ domain, respectively $(u, \hat{u}), (u, u_r) \in Y \times Y$, and so, the GAN loss was added, both for shadow adding/removal step and the image recovery step. So, the $L^f_{GAN}(\hat{u}, u), L^f_{GAN}(u, u_r)$ represent the *shadow free generation GAN loss* and the *shadow free recovery GAN loss*, and, respectively $L^s_{GAN}(\hat{v}, v), L^s_{GAN}(v_r, v)$ the *shadow generation GAN loss* and *shadow recovery GAN loss*.

As the images u and u_r should be identical in terms of pixel-wise properties and contents, as well as the v and v_r pair, the L1 loss was introduced (as $L_{pix}(u, u_r), L_{pix}(v, v_r)$). As the recovered results should be similar to the inputs in terms of content, color and style, the perceptual loss ($L_{perceptual}(u, u_r)$, $L_{perceptual}(v, v_r)$) was introduced.

The cycle consistency imposed will imply that the shadow mask will not change along the transformations computed, and so, the terms $L_{mask}(\hat{m^f}, m_r^f)$, $L_{mask}(\hat{m^s}, m_r^s)$ will count the difference between the masks computed after the forward step and the ones produced by the recovered images.

As the model will benefit from a higher quality representation, when paired images are used for training, the pixel-wise L1 loss between the ground-truth shadow free image and the synthetic one, $L_{pix}(\hat{u}, u)$, and also the L1 mask loss $L_{mask}(m, \hat{m^s})$ were introduced, to decrease the time needed for the model to learn a realistic mapping. The same reasoning can be applied for the shadow masks produced along the cycle, and so, in the paired setting, the β_1 and β_2 were introduced, to accelerate the convergence in the training procedure, by minimizing the differences between partial results and the provided ground truth.

As the shadow is irrelevant when trying to semantically describe the image, the high-level features learnt by a CNN are to be the same for both the shadow free image u and the shadow image v. So along the cycle, regardless the type of the generator, with the input image x should have approximately the same semantically relevant content as the image $G(x)$. So, for the both mappings, G_f and G_s, the content loss terms were introduced, as $L_{content}(u, \hat{v}), L_{content}(v, \hat{u})$. The presented terms will form the training objective for the generators, as a linear combination. The coefficients chosen will have an impact in the training process, as the loss function will be dominated by the greatest magnitude, and so, the values had to be chosen after observing the magnitude and the decrease rate for each of the terms.

1.3. Unpaired vs. paired settings

1.3.1 Paired training

We refer to the *paired training* as the procedure of training the model using all the provided information for the training images in the ISTD dataset [3]. So, in the training procedure, the pair (u, v) will be a pair of representations of the same scene, with one of the images affected by the shadow. The images u and v will also be aligned with the ground truth shadow mask m, and so, the model will receive also the true localization information. The problem can be solved as a regression task, but the highly limited size of the training set will affect the generalization ability of the model.

For training the discriminators the image pair (u, v) will be used as follows: the discriminator will ask its pair generator for a more realistic version of the mapped input image, so, the output for the discriminator is expected to be close to all-1 matrix J for a synthetic image equal to the expected one, and close to all-zero for a pair of images which has the synthetic generated image the same as the input (they can not be in different domains). The $L_{mse}(x, y)$ is the mean squared error defined for the x, y input.

This strategy, of penalizing the pixel-wise similarity between the input image and the output will stimulate the generators to learn a realistic mapping. As the information provided for the model is complete, the β_1 and β_2 parameters can be used to accelerate the convergence of the training procedure.

1.3.2 Unpaired training

In the unpaired training setting (employing self-supervised learning), the images used along the cycles will not represent the same scene, such that conditioning on the expected shadow free image can not be used, in order to guide the training procedure.

At each step in training, two images will be randomly sampled, without replacement, and both the synthetic and the recovered results are computed. The ability of the model to detect and restore the shadow image has to come from imposing the cycle consistency constraint, where the recovered images are, asymptotically, converging to the input images used, when the true mappings are learned by the corresponding generators.

The *shadow adder generator* needs also the shadow mask, to produce a realistic result (as the differences between the input image and the expected output are not so significant for the mask to be learnt for each of the samples used for training). In the unpaired setting, the derivation of the ground-truth mask with any of the results available along the cycles is impossible, as the images represent different scenes. So, the model can use just a synthetic mask. A binary shadow mask will be computed as the binarization of the difference between the synthetic shadow free image and the ground-truth shadow image used as input for the *shadow remover generator*, pushed into a memory buffer and, during training, synthetic shadow masks are sampled in order to perform the forward shadow addition procedure.

In Figure 3 we depict intermediary results from the training cycle involving shadow addition and removal and recovery of shadow and shadow-free images and masks. The task is getting more difficult to solve, and so, to help the model generalize better, the synthetic shadow masks were kept in a temporary buffer, of a fixed size, when, during the training, new samples are replacing the old ones. The masks produced this way will be used in the shadow addition of *forward step*, where a randomly sampled mask will be used for the synthetic shadow image computation. For the recovery step, the synthetic mask $\hat{m^f} = Bin(\hat{u} - v)$ will be used. This random sampling of the masks will help the model generalize better, as the number of training combinations that can be produced is definitely higher, compared to the limited set of scenes from the training set.

$$L^s_{GAN}(u,v) = \frac{1}{2}(L_{mse}(J, D_s(G_s(u), v)) + L_{mse}(O, D_s(G_s(u), w))),\ \forall w \notin X \tag{1}$$

$$L^f_{GAN}(u,v) = \frac{1}{2}(L_{mse}(J, D_f(G_f(v), u)) + L_{mse}(O, D_f(G_f(v), w))),\ \forall w \notin Y \tag{2}$$

For the discriminators training, two additional image buffers were added, following the same reasoning of creating variability in the training samples. The same memory queue was used, to benefit from the improvement of the results, in terms of quality, as the training procedure continues. So, when looking at the equations (1), (2), we can observe that, for the negative example in the training of each of the discriminators, a sample from the other domain is needed. Fixing this sample conveniently to u or v image is sub-optimal, because a source of variability in the sample set can improve the performance of the discriminators, enabling the generators to produce better results. So, for the shadow domain X and shadow free domain Y, two image buffers will be added, randomly sampling the w image as stated in the equations (1), (2).

2. Ablative study

2.1. Training: influence of data and number of epochs

2.1.1 Training data

To test the influence of the training number of samples in the generalization ability of the model, we created the following setup. We used the architecture for the unpaired training version of the model and gradually reduced the number of training samples used from the ISTD dataset [3], and then, performed a validation step to measure the ability of the model to generalize on the ISTD testing samples. The results are shown in Figure 4. As expected more training data leads to better performance (in RMSE terms) with fewer training epochs and the training is more stable.

2.1.2 Number of epochs

Here we analyze the behaviour of the model throughout the training procedure. The decreasing trend can be observed in both Figure 5 and Figure 6 for both RMSE and LPIPS [4]. Wisely decreasing the learning rate, the performance of the model can be improved, both in terms of pixel-wise and perceptual loss functions.

The better performance of the unpaired setting, both in terms of RMSE and LPIPS, can be explained by the better

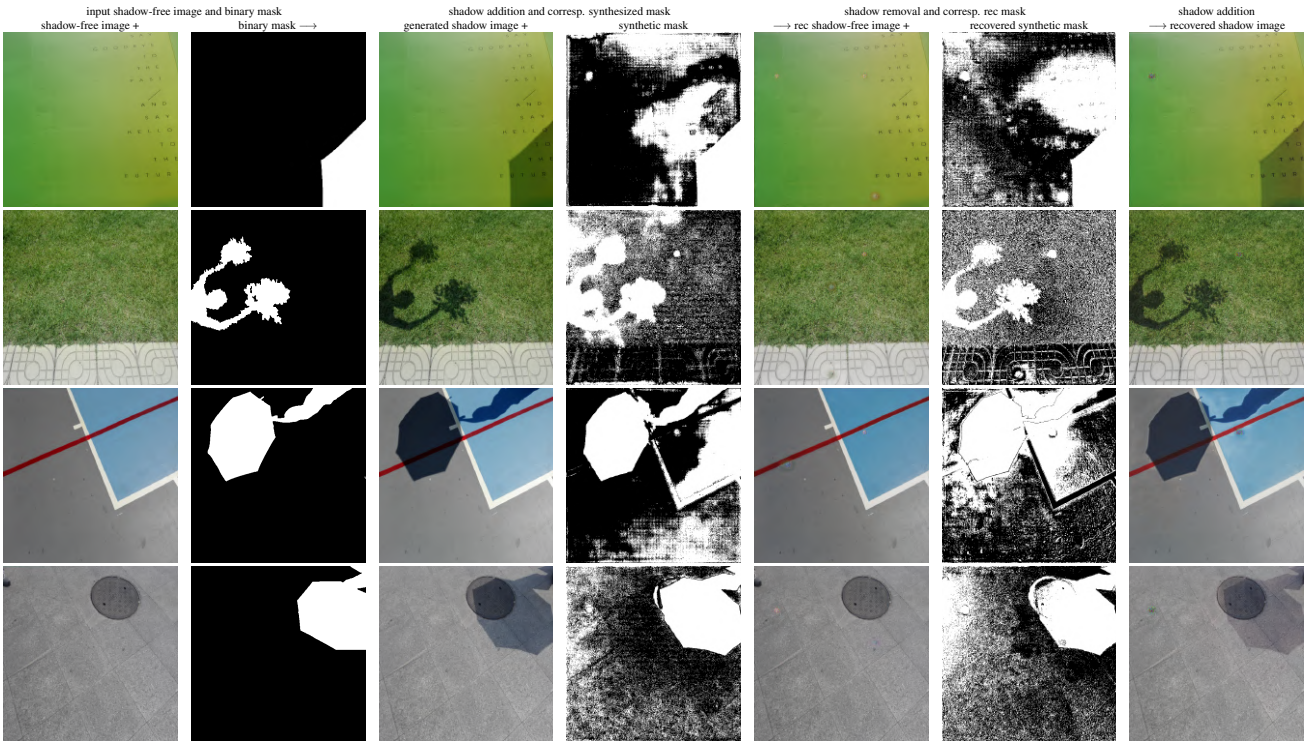


Figure 3: Visualized circles of shadow addition, removal, and reconstruction of shadow free, shadow mask and shadow images in our self-supervised approach (unpaired training). Shadow adder takes as input a shadow free image and a random binary shadow mask and synthesizes a realistic shadow image (the difference to the input provides the synthetic shadow mask). Shadow removal takes as input a shadow image and synthesizes a shadow free image (the difference to the image provides the recovered shadow mask). All the synthesized/recovered masks are depicted after binarization.

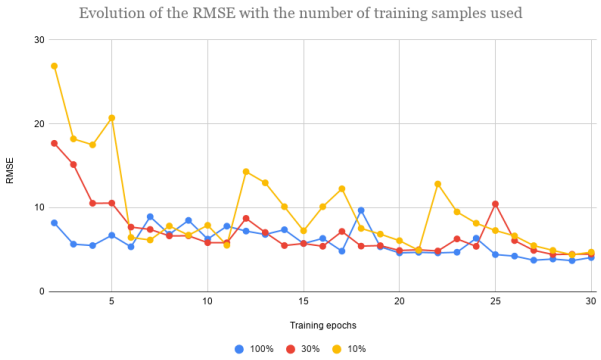


Figure 4: Accuracy (RMSE) vs. number of training epochs and amount of training samples used from the ISTD dataset. Here the results are reported on ISTD test images.

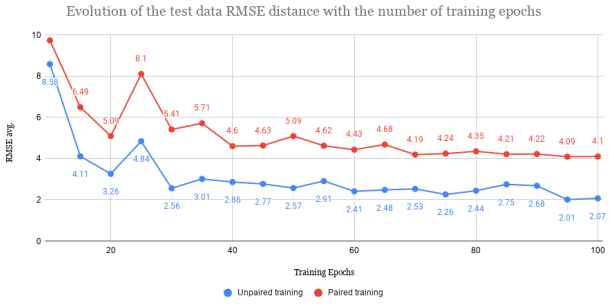


Figure 5: Evolution of the RMSE pixel-wise error for the recovered images with the number of training epochs employed for our models in unpaired and paired training settings.

generalization ability of the model, enhanced by the random sampling for the (binary) shadow masks for generators training and the negative samples during the training of the discriminators.

2.1.3 Data augmentation

We proposed a training strategy based on random transformations. In detail, for every sample in the test set, the model will be fed with the original sample and another k versions of the original images, where a random rotation and a ran-

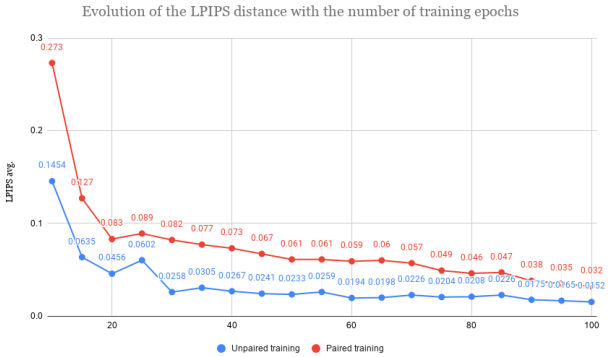


Figure 6: Evolution of the LPIPS perceptual quality loss for the recovered images with the number of training epochs employed for our models in unpaired and paired training settings.

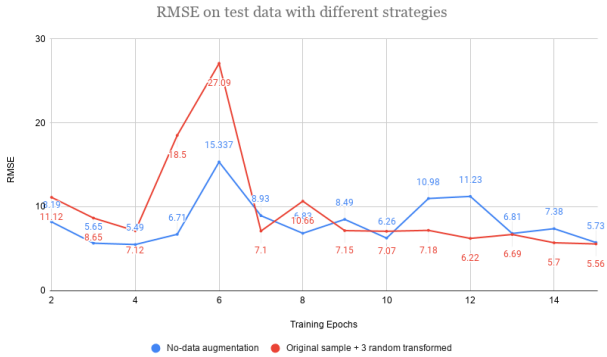


Figure 7: Data augmentation strategy influence on accuracy (RMSE).

dom flip are performed. The losses involved in the training objective for the generators will be computed as the average of that particular loss for each of the versions used in training. A reduction in terms of RMSE can be observed, but, as the cycle has to be traversed $k+1$ times, the training time will increase.

For the discriminators training process, the synthetic shadow/shadow free images used as inputs were also transformed, and, the discriminators were trained with the original image, and its $90°$, $180°$, $270°$ rotated versions. The GAN loss, for the positive pair, will be computed, as well as in the previous case, as average over the prediction error for each variation of the synthetic image. As the model for the discriminator is significantly more simple, the effect in training time per epoch is negligible.

In Figure 7, we present the influence in the evolution in terms of RMSE distance between the ground truth images and the recovered ones, and, after a natural transition in the early epochs, produced by the higher number of examples used in training, the used strategy helps the model generalize better on the test set in the later stages, with a much more stable decrease.

2.2. Design and losses

2.2.1 Mask

The shadow mask is a set of binary inputs $m_{i,j}$ that tells, for a shadow affected image, if the particular pixel $I^s_{i,j}$ is shadow affected or not. As every shadow free image can be written as a linear combination of a shadow image and a compensation image that will increase the illumination for the shadow affected pixels, the shadow mask provides information about the region of the image that the generator has to change in order to perform a valid translation. Also, the color compensation is necessary to achieve high-quality results, and so, the learning of a shadow-to-free mapping is expected to perform better.

As our model uses normalized data in the [-1, 1] interval, the particular mask computed after a particular forward step is a set of binary values $m_{i,j} \in \{-1, 1\}$, where a 1 value means that the corresponding pixel is shadow affected. We compute the mask as the thresholding of the difference between the synthetic shadow free image and the true shadow affected image. The value for the threshold was set as the median value of the grayscale representation mapped image difference.

2.2.2 Losses

In Figure 9, the evolution of the RMSE loss is reported for the initial stage of the training procedure for two different settings. For the first setting, we considered the unpaired training strategy with the coefficient $\gamma_2 = 10$, and then, we set $\gamma_2 = 0$. As the mapping learnt would not be constrained by the content of the synthetically generated image, a drop in performance can be observed.

So, we can conclude that the control over the semantically relevant information, extracted with the high-level features given by the Perceptual Loss Module, provides sufficient control after the *forward step* of the cycle, such that a better intermediate mapping can be learnt such that the recovered image will be better in terms of pixel properties and semantic content.

In the Table 4 from the main document, we summarize the procedure of analyzing the effect of each of the parameters involved in the training objective, by comparing the results produced by the model after the early stage (15 epochs) of its training. As it can easily be seen, the standard set of parameters is the best trade-off to achieve high quality results in terms of pixel-wise and also, perceptual error functions. For the unpaired setting, the $\gamma_1 and \gamma_2$ parameters are the most important, as, in order to decide about

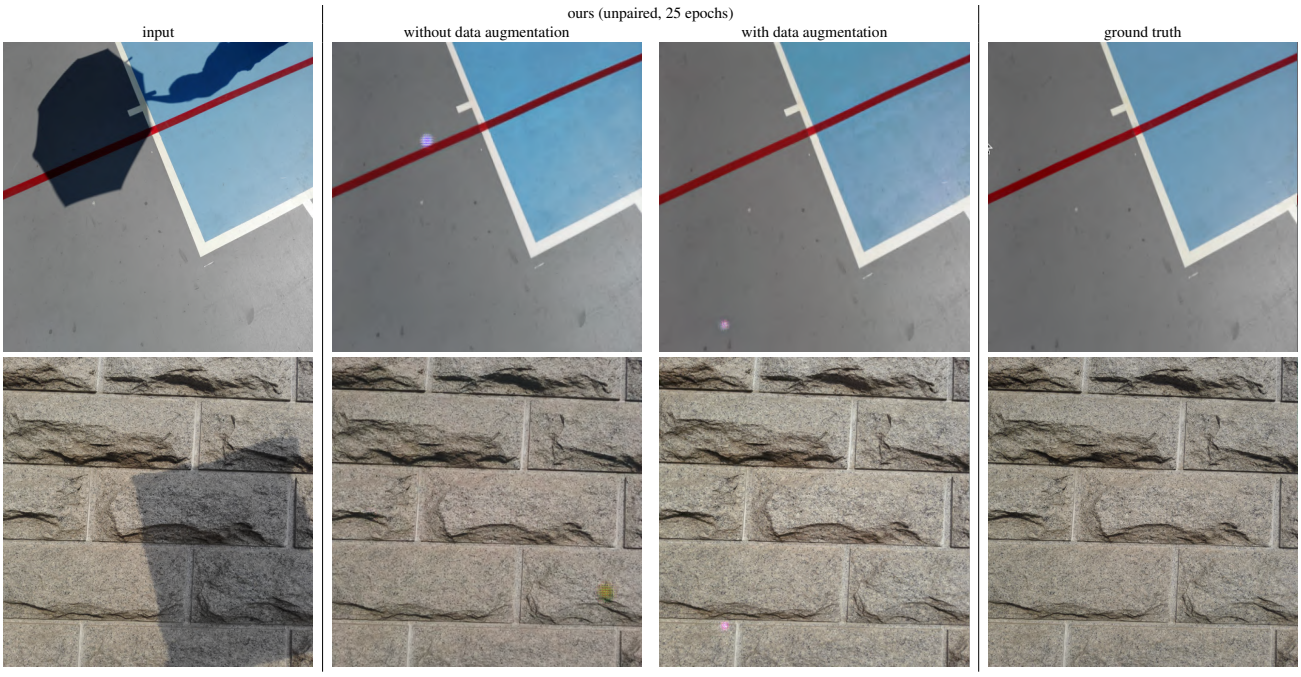


Figure 8: Data augmentation effect on the performance enhancement. Inputs were randomly sampled and the results generated using the unpaired setting, with and without data augmentation and trained for 25 epochs.

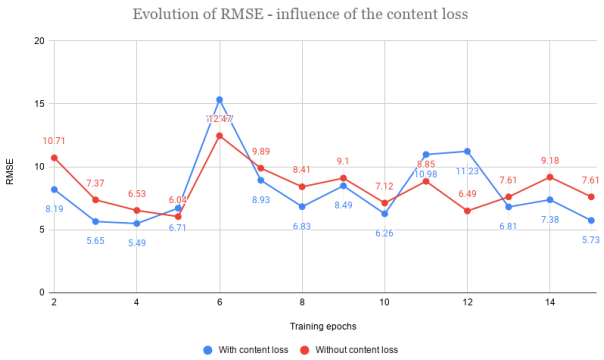


Figure 9: Influence of the content loss in the evolution of RMSE

the output, the generators can benefit from the content constraint and the generalization of the discriminator, to produce a better result.

The γ_3 parameter that controls the cycle constraint contribution in the loss function, is important in order to produce good quality results in terms of pixel-wise properties. The better results produced after suppressing the perceptual loss contribution in the training objective (by setting γ_4 to 0), can be explained by the reduction of the artifacts effect in the produced images. However, even if the pixel-wise performance of the model seems good-enough, the results in terms of perceptual quality (LPIPS) are not as good as the pixel-wise loss measurements, as we expected, because we used the perceptual loss module in order to overcome the difficulties coming from the illumination and semantic inconsistencies of the training set.

The γ_5 parameter, that controls the contribution of the mask consistency constraint along the transformations computed, is also, naturally, very important in order to produce good results in terms of the semantic content represented. The shadow adder generator needs valid information about the shadow mask, and, as it is computed using the synthetic output of the shadow remover generator, both generators will benefit from enforcing such a constraint.

Removal of perceptual loss ($\gamma_4 = 0$) and/or mask loss ($\gamma_5 = 0$) show improvement in terms of pixel-wise accuracy after 15 epochs of training. However, as discussed in the paper, our target is perceptual, and in terms of perceptual quality measured by LPIPS, we can observe the degradation with respect to the default setting which includes both losses. A visual inspection of the obtained results aligns with LPIPS, there are significantly fewer visual artifacts in the shadow free images recovered with our method employing both the perceptual and the mask losses.

3. Additional visual results

Given the known inconsistencies int the ISTD dataset, in [2], the authors proposed a method aiming to compen-

sate for the difference in the global illumination between the shadow affected frame and the shadow free image, by exploiting the statistics over the unaffected regions in the input shadow affected image. So, the ISTD+ dataset was created, by applying the local correction method over the original images from the ISTD dataset. In the Figure 10, we compare our method against SP+M-Net [2], providing samples produced by both methods.

In Figure 11 we provide several results[1] on USR test images for our self-supervised approach (unpaired training) and for the Mask Shadow GAN [1] approach. Mask Shadow GAN generally produces significantly more visual artifacts and has difficulties to remove the shadows in these images.

The discriminators are expected to learn the characteristics of both shadow and shadow free domains. By observing the results in the Figure 11, we can conclude that the proposed strategy for discriminators training, coupled with the data augmentation strategies we deployed, are offering a better generalization ability. The usage of a perceptual loss will enable the conditioning on the content observed in the input shadow affected image, that is going to provide another degree of control in the training procedure, resulting in a faster decrease rate.

This enables the representation of the samples in a convenient latent space, such that, using the synthetic partial results $(\hat{u}, \hat{v})$, the quality of the reconstructed images will be enhanced, both in fidelity loss value and perceptual score.

References

[1] Xiaowei Hu, Yitong Jiang, Chi-Wing Fu, and Pheng-Ann Heng. Mask-ShadowGAN: Learning to remove shadows from unpaired data. In *ICCV*, 2019. 1, 8, 10

[2] Hieu Le and Dimitris Samaras. Shadow removal via shadow image decomposition. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 7, 8, 9

[3] Jifeng Wang, Xiang Li, and Jian Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1788–1797, 2018. 3, 4

[4] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[1] Models and results will be made public on the web page of the project after the paper is accepted.

Figure 10: Visual results on the ISTD+ dataset, comparing our paired setting to SP+M-Net model[2]. Better zoom in on screen.

Figure 11: Visual results on USR test images for our self-supervised model trained with unpaired images (*top*) and for Mask Shadow GAN [1] (*bottom*). Note that the results are provided for reference, were randomly sampled and are not corresponding to the same input shadow image. Better zoom in on screen.