ShadowNet

Eli Kaminsky and Michael Werman(✉)

Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel
michael.werman@mail.huji.ac.il

Abstract. Finding shadows in images is useful for many applications, such as white balance, shadow removal, or obstacle detection for autonomous vehicles. Shadow segmentation has been investigated both by classical computer vision and machine learning methods. In this paper, we propose a simple Convolutional-Neural-Net (CNN) running on a PC-GPU to semantically segment shadowed regions in an image. To this end, we generated a synthetic set of shadow objects, which we projected onto hundreds of shadow-less images in order to create a labeled training set. Furthermore, we suggest a novel loss function that can be tuned to balance runtime and accuracy. We argue that the combination of a synthetic training set, a simple CNN model, and loss function designed for semantic segmentation, are sufficient for semantic segmentation of shadows, especially in outdoor scenes.

Keywords: Shadow detection · IoU - Intersection over Union
Basin-loss function

1 Introduction

This paper presents a CNN to segment shadows in an image. To that end we created a synthetic set of shadows and randomly added them to hundreds of hand picked shadow-less images. We show that training a CNN on a synthetic set of shadows with a novel loss function designed for semantic segmentation, yields satisfactory segmentation results on real images.

Past [1,6,10,12] and contemporary work [2,7–9,11] on shadow segmentation relied on manual or semi-automated labeling of shadowed pixels in an image in order to create the training, validation and test sets. Hand labeling shadows is very time consuming, and of course, not all-possible shadow shapes and intensities can be found and labeled for supervised learning. Using synthetic shadows we can simulate an unbounded range of intensities and shapes.

Resizing or reshaping an image or synthetic shadow is possible since shadows are often cast from an object not present in the image. Synthetic shadows from the test set are shown in Fig. 1. Images on the left are with cast shadows, images in the center are the ground truth shadow segmentation, and the images on the right are the CNN predicted segmentations.

This research was supported by the Israel Science Foundation and by the Israel Ministry of Science and Technology.

A. Campilho et al. (Eds.): ICIAR 2018, LNCS 10882, pp. 336–344, 2018.
https://doi.org/10.1007/978-3-319-93000-8_38

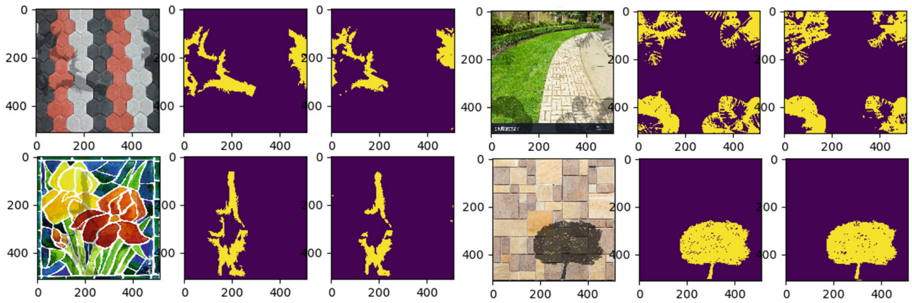

Fig. 1. In each triplet, scene with cast shadows, ground truth shadow segmentation, and the predicted segmentation

2 CNN Model Structure

In our dataset, shadows appear both in low frequencies varying slowly across the scene or as very high frequency intensity changes. Thus, we should apply different sized kernels in order to extract their features and segment them. As large kernels are computationally expensive, we can replace such kernels with cascaded smaller kernels or run them on a subsampled image. Convolving with an $n \times n$ kernel on an image resized by 0.5, has the effect of running a $2n \times 2n$ kernel on the original image. As subsampled images may lose small artifacts or high frequency features at subsampling, we ought to keep extracted features for high and low frequencies of pre-subsampling layer. The U-Net architecture [3] with multi-scaling tensors at several resolutions fulfills our requirements and we based our model on it.

As we only semantically segment shadows we did not require the full U-Net architecture. We removed the last subsampled layer and reduced the base number of convolutional kernels from 64 to 24, Fig. 2 depicts our model.

3 Loss Function

The error, or loss, between the ground truth and the predicted image is often taken to be the pixel-wise Root-Mean-Square-Error, $Loss_{RMSE}$. Using $Loss_{RMSE}$ the model converged fast. However, since shadows are often not a dominant segment of the scene this resulted in a model that ignored small shadow segments.

For semantic segmentation, it is often customary to set the loss function in terms of IoU (intersection of union).

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{1}$$

With the loss function being:

$$Loss_{IoU} = 1 - IoU \tag{2}$$

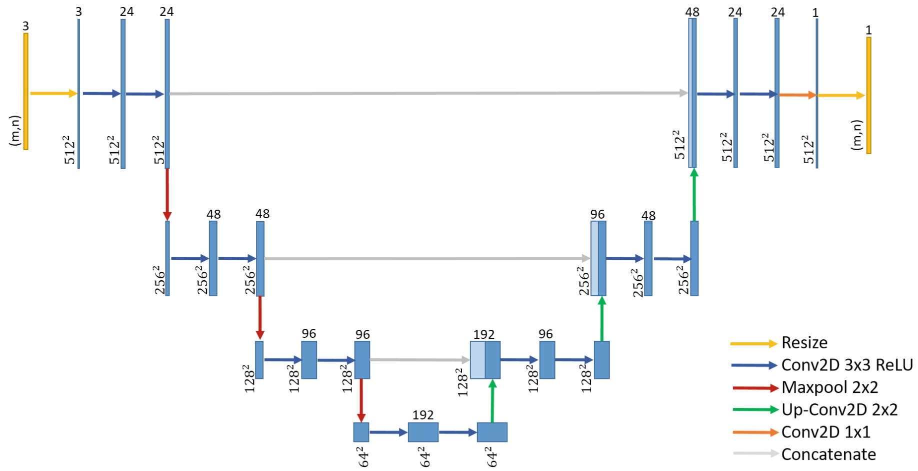

Fig. 2. A U-Net based architecture for shadow segmentation.

$Loss_{IoU}$, is indifferent to the shadow-less part of the scene. This is especially important in order not to miss small shadows segments which are often ignored by a pixel-wise loss. However, minimization of $Loss_{IoU}$ converges very slowly compared to $Loss_{RMSE}$, as it gives an equal weight to small shadow segments and to large ones in an image.

In order utilize both of their relative advantages - fast minimization, $Loss_{RMSE}$, and accuracy, $Loss_{IoU}$, we set the loss function to a combination of them. The simplest combination being linear:

$$Loss_{total} = \alpha Loss_{RMSE} + (1 - \alpha) Loss_{IoU} \tag{3}$$

In Eq. 3, $Loss_{total}$ is a plane. Gradients pointing to the most significant change in the loss function are parallel (slope of the plane), so gradients have no preference and do not converge to a single minimum at the origin. Therefore, convergence might be slow and accurate in case gradients are close to the $Loss_{IoU}$ axis, or faster and less accurate in case gradients are closer to the $Loss_{RMSE}$ axis.

For a more stable form of a loss function we used a quadratic combination of $Loss_{RMSE}$ and $Loss_{IoU}$:

$$Loss_{total} = Loss^2_{RMSE} + Loss^2_{IoU} \tag{4}$$

Unlike Eq. 3, it is impossible to minimize only one component of the loss function without the other in training, since gradients have radial symmetry and all point to a single minimum at the origin. Figure 3 shows contours and gradients for the loss functions in Eqs. 3 and 4.

Nonetheless, while training on our sets, there was a "pull" of the gradients towards the $Loss_{RMSE}$ axis, since it was easier to minimize $Loss_{RMSE}$ than

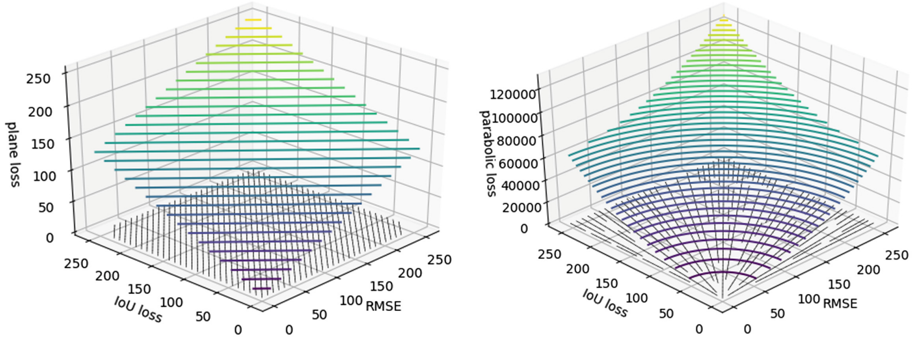

Fig. 3. 3D contours of plane loss and parabolic loss of Eqs. 3 and 4. For plane loss, projection of parallel gradients on the $Loss_{IoU}$-$Loss_{RMSE}$ plane. For parabolic loss, projection of radial gradients on the $Loss_{IoU}$-$Loss_{RMSE}$ plane.

$Loss_{IoU}$ neglecting many small shadows segments. Therefore, we chose a new loss function that forces the gradients towards the $Loss_{IoU}$ axis to increase accuracy. We required the loss function to have a basin shape with a single minimum and a basin line through which we could force the gradients to minimize the function to a preferred axis. In Eq. 4, if we divide the term $Loss^2_{RMSE}$ by $Loss_{IoU}$ and divide the term $Loss^2_{IoU}$ by $Loss_{RMSE}$, we penalize $Loss_{total}$ if one of the terms becomes dominant, as $Loss_{total}$ increases. The penalty is minimized when $Loss_{RMSE} \approx Loss_{IoU}$. The following equation achieves just that:

$$Loss_{total} = \frac{\alpha Loss^2_{RMSE}}{Loss_{IoU}} + \frac{Loss^2_{IoU}}{\alpha Loss_{RMSE}} \tag{5}$$

The hyper-parameter α, tunes the slope of the basin line of the loss function, $Loss_{total}$, which roughly represents the line $y = \alpha x$. We denote Eq. 5 as "basin-loss", illustrated in Fig. 4. Note that since $Loss_{RMSE}$ and $Loss_{IoU}$ are differentiable [4,5], Eq. 5 is also differentiable and can be used for back-propagation.

4 Training and Results

For our dataset [13], we chose 580 shadow-less images from the web, and with Photoshop created 100 shadow objects. As we used a PC-GPU, to make the most of the GPU memory we ran our model on mini-batches of size 5 for 30 iterations per epoch. We resized input images to 512 × 512 × 3. For each image we randomly chose a shadow object, randomly blurred it to soften its edges, randomly set its intensity, and randomly chose an affine transformation to resize, skew, and rotate it. Finally, we added the shadow object to the image, see Fig. 1.

Our sets of images were divided $80\% - 10\% - 10\%$ for training, validation, and test, ($\alpha = 1.1$). There are two measures of accuracy; first – pixel accuracy, or percentage of pixels that are equal to 1 or 0, when comparing predicted shadow with ground truth, second – percentage of intersection over union, or

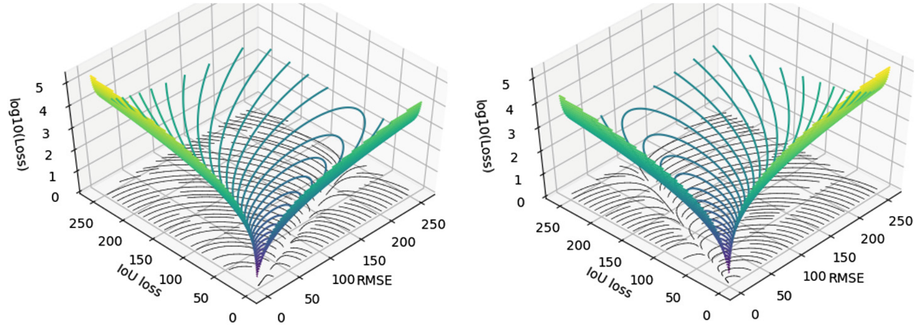

Fig. 4. Contours of basin-loss of Eq. 5, in log scale. Left: basin-loss with $\alpha = 0.3$, with a projection of the gradients on the $Loss_{IoU}$-$Loss_{RMSE}$ plane. The "pull" of the gradients is towards minimizing IoU loss. Right: basin-loss function with $\alpha = 3$, with projection of the gradients on the $Loss_{IoU}$-$Loss_{RMSE}$. The "pull" of gradients is towards minimizing RMSE loss.

IoU accuracy, of predicted shadow with ground truth. The training set converged to 96% pixel-accuracy and validation converged to 95%. For IoU accuracy, the training set converged to 82% and validation to 80%.

There is a trade-off in time-to-converge and IoU accuracy based on α. Training with $\alpha \leq 1$ caused the model to converge too slowly. If the basin-line is such that the model is forced to compensate for $Loss_{IoU}$ more than for $Loss_{RMSE}$, model convergence becomes very slow. Figure 5 shows model training results for different values of α in $Loss_{IoU}$ and $Loss_{RMSE}$.

Note in Fig. 5, that the difference in loss between the axes increased as α grows. Also setting α too large, say $\alpha = 3$, degrades IoU performance.

We used the 'Adadelta' optimization algorithm with a learning rate reduction of 90% when there is no progress. The effect of improvement in accuracy by learning rate reduction is visible in Fig. 5 (top two images), at epoch 530.

Figure 6 shows examples with real shadows we tested the shadow segmentation on and Fig. 7 shows how the CNN finds small shadows.

5 Comparison with Previous Work

We found our work to be a little better than [1] in terms of shadow segmentation. We ran our model on the test set and ground truth of test set of [1]. To improve test accuracy, we performed Otsu thresholding on our outputs and achieved 90% pixel accuracy which is 1% improvement compared to [1], and with 72% IoU accuracy - no such parameter to compare to. We could not run the code of [1] to test our samples. It is also important to note that ground truth given for these images is often incorrect both in marking non-shadows as shadows and in missing parts of the shadow. This results in a significant accuracy loss, especially in areas where our model had segmented a shadow correctly and the ground truth did not. Figure 8 shows examples.

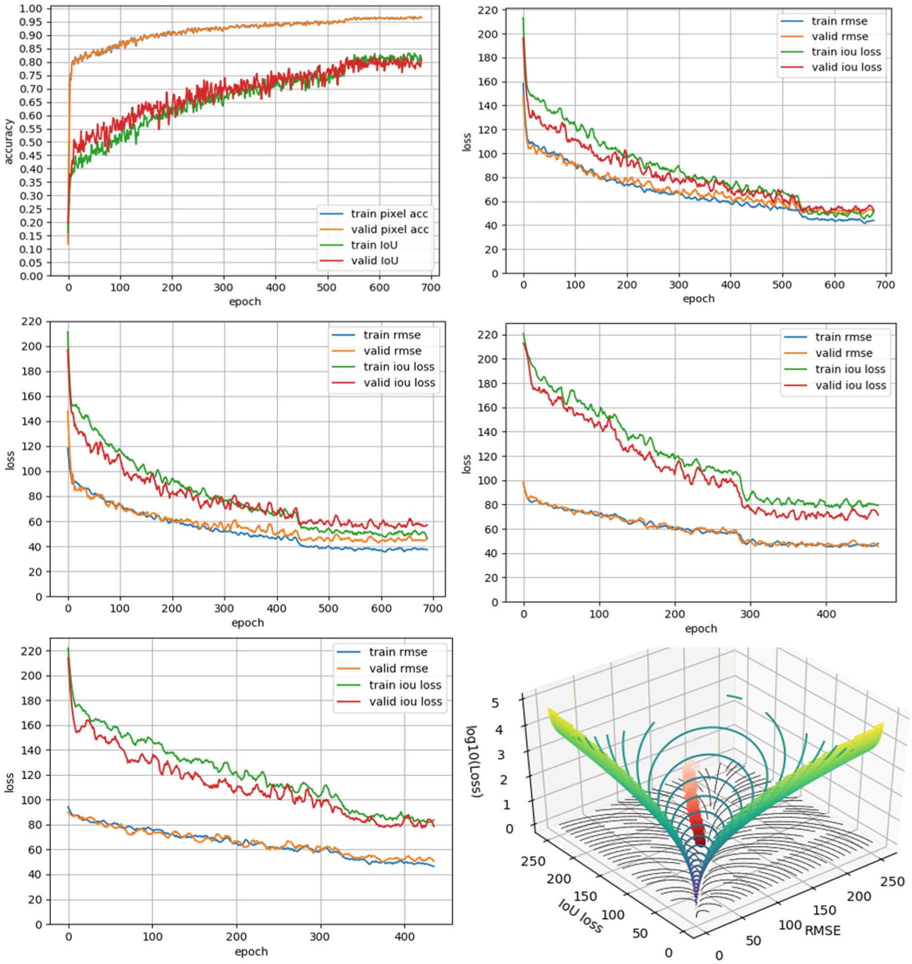

Fig. 5. Comparing training and total loss between Eqs. 4 and 5. Top left: Train/validation pixel-accuracy/IoU-accuracy for Eq. 5 with $\alpha = 1.1$. Top right: losses for basin-function with $\alpha = 1.1$. Center left: losses for basin-function with $\alpha = 2$. Center right: losses for basin-function with $\alpha = 3$. Bottom left: losses for Eq. 4, $Loss_{RMSE}$, with $Loss_{IoU}$ only measured. Bottom right: $Loss_{total}$ values illustrated on basin defined by Eq. 5 with $\alpha = 1.1$. Loss defined by Eq. 5 (basin loss) performs better for $Loss_{IoU}$.

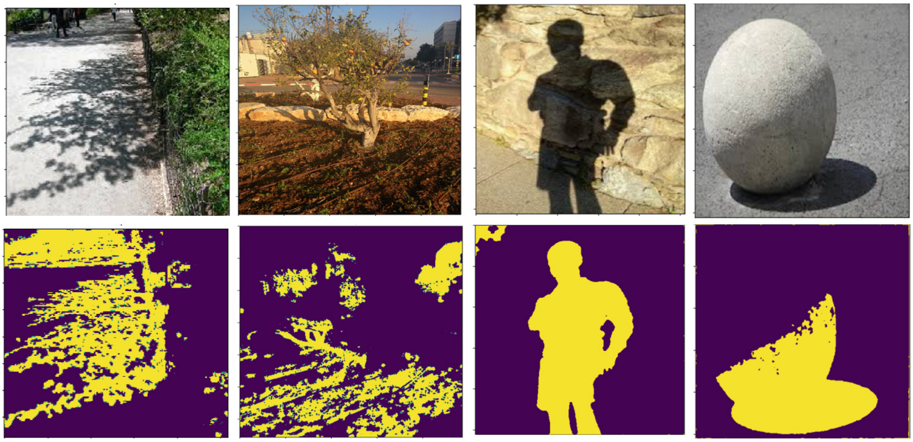

Fig. 6. Top, outdoor images with shadows. Bottom, shadow segment.

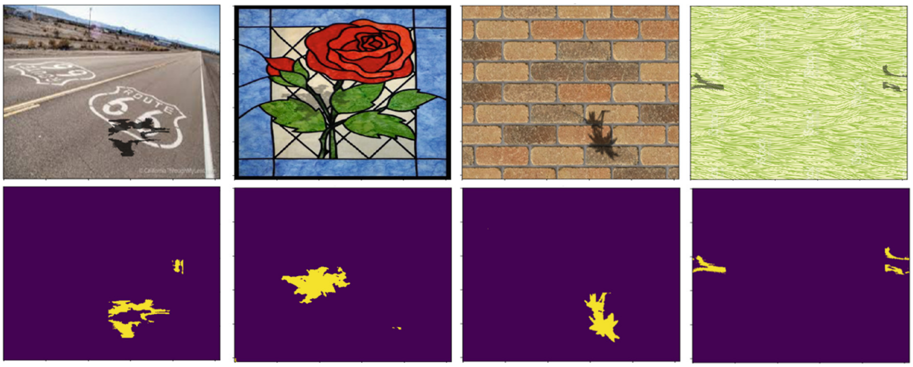

Fig. 7. Top, images with small shadows. Bottom, shadow segment.

We were unsuccessful in applying the code of [2] due to major compilation problems. However, we present samples of our model on some of their images. Images we downloaded from [2] were mostly of indoor with lots of soft shadows and no clear edges, nevertheless, shadow segmentations were successful, as shown in Fig. 9.

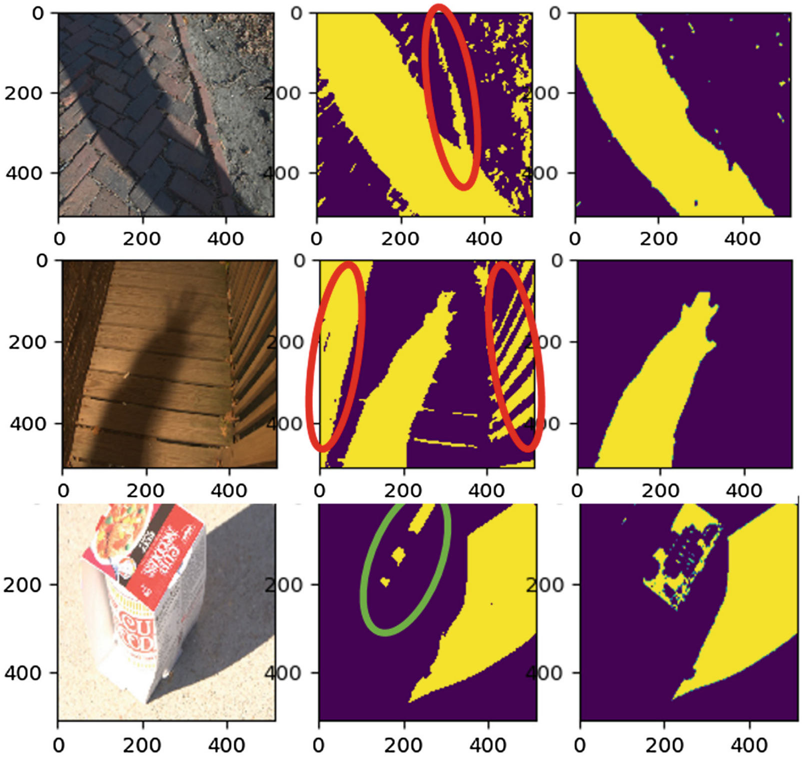

Fig. 8. Left: input images, center: our shadow segmentations and right the ground truth given in [1]. Misclassified ground truth is marked with red ellipses. Note that the given ground truth is not always reliable. (Color figure online)

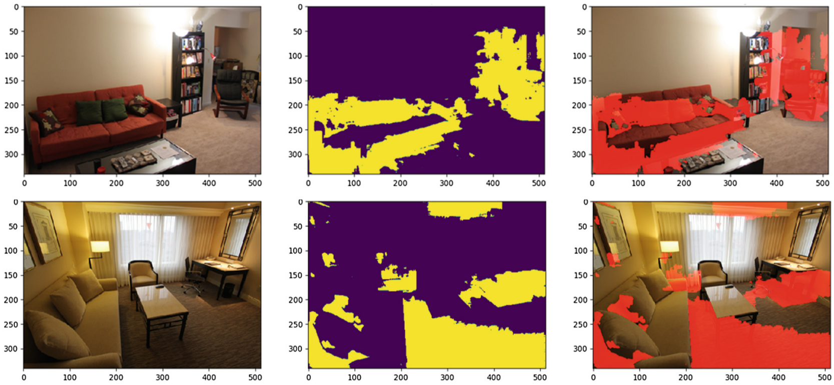

Fig. 9. Samples from [2]. Left, inputs, center our segmentation, and right are hand-marked shadows.

6 Conclusion

We showed that a small CNN trained on a small synthetic set of shadows, a small set of shadow-less images, and a loss function designed to minimize runtime and loss of both Root-Mean-Square-Error and Intersection-over-Union-Loss can give good shadow segmentations.

More generally, our work illustrates the importance of choosing a proper loss function for semantic segmentation. Since even if with an abundance of data a network can only be trained as well as its loss function allows and that designing the gradients' path can improve the overall accuracy of a network.

References

1. Guo, R., Dai, Q., Hoiem, D.: Single-image shadow detection and removal using paired regions. In: CVPR 2011 (2011)
2. Kovacs, B., Bell, S., Snavely, N., Bala, K.: Shading Annotations in the Wild. arXiv:1705.01156 (2017)
3. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
4. Optimizing IoU Semantic Segmentation. http://angusg.com/writing/2016/12/28/optimizing-iou-semantic-segmentation.html/
5. Rahman, M.A., Wang, Y.: Optimizing intersection-over-union in deep neural networks for image segmentation. In: Bebis, G., et al. (eds.) ISVC 2016. LNCS, vol. 10072, pp. 234–244. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50835-1_22
6. Blajovici, C., Kiss, P.J., Bonus, Z., Varga, L.: Shadow detection and removal from a single image. In: SSIP 2011 (2011)
7. Nguyen, V., Vicente, T.F.Y., Zhao, M., Hoai, M., Samaras, D.: Shadow detection with conditional generative adversarial networks. In: ICCV 2017 (2017)
8. Russell, M., Zou, J.J., Fang, G.: An evaluation of moving shadow detection techniques. Comput. Vis. Med. **2**(3), 195–217 (2016)
9. Vicente, T.F.Y., Hou, L., Yu, C.-P., Hoai, M., Samaras, D.: Large-scale training of shadow detectors with noisily-annotated shadow examples. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 816–832. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_49
10. Zhu, J., Samuel, K.G., Masood, S.Z., Tappen, M.F.: Learning to recognize shadows in monochromatic natural images. In: CVPR 2010 (2010)
11. Khan, S.H., Bennamoun, M., Sohel, F., Togneri, R.: Automatic shadow detection and removal from a single image. In: PAMI 2016 (2016)
12. Lalonde, J.-F., Efros, A.A., Narasimhan, S.G.: Detecting ground shadows in outdoor consumer photographs. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6312, pp. 322–335. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15552-9_24
13. Dataset and Source Code. http://www.cs.huji.ac.il/~werman//