Shadow Detection Using Robust Texture Learning

Tianxiang Pan[1]
ptx9363@gmail.com
Bin Wang[1]
wangbins@tsinghua.edu.cn
Guiguang Ding[1]
dinggg@tsinghua.edu.cn
Junhai Yong[2]
yongjh@tsinghua.edu.cn

[1] School of Software
Tsinghua University
Beijing, China

[2] Beijing National Research Center for
Information Science and Technolog
Beijing, China

Abstract

In this paper, we propose a simple but effective texture learning network to improve the shadow detection. It is based on an interesting observation. We found shallow networks perform better than deeper networks on the task of shadows detection. It suggests shadows may be more correlated with basic textures, like harder colors, than deep semantic information. Because shadow and non-shadow areas on the same ground may share similar textures, we need to find the best boundary between them. In this paper, we propose a novel hard-negative mining module to extract different texture patches that are supposed to be the hardest patches to distinguish between shadows and non-shadows. We also design an end-to-end trainable structure to integrate the texture features with deep semantic fully convolutional networks. The texture learning module is lightweight and the end-to-end structure makes it possible for our module to help other detection tasks. Results on a variety of ablation experiments confirm the improvement brought by our texture learning module. Moreover, the final detection model achieves a state-of-art detection accuracy on most benchmarks.

1 Introduction

Shadows, as one essential aspect of visual images, play key roles in computer vision and image understanding. Shadows in images may cause distraction and difficulty into the majority of computer vision tasks, including classification, object detection, and semantic segmentation. Surfaces with shadow can be hard to distinguish even for humans. Shadow-free images are reasonably more effective for classification and object detection. Meanwhile, precise shadow detection can provide luminance information and scene understanding of specific applications.

Early works on shadow detection mainly focused on physical feature of illumination and colors [8][10][3][4]. These methods worked poorly on difficult natural images because of their strict illumination assumptions. In recent years, motivated by the advancement of deep learning models, several methods were proposed based on three mainstream structures: CNN(Convolution Neural Network), FCN(Fully Convolutional Network) and GAN(Generative Adversarial Networks). CNN is firstly introduced by Khan et al. [7] as a shadow edge classifier to detect shadows. Vicente et al. [14] take advantage of FCN and attempted to integrate semantic prior into shadow detection. They also collected a new large shadow dataset for a fair comparison between data-driven methods. Nguyen et al. [11] extend conditional GAN as a generator to generate shadow mask. A sensitivity parameter is proposed by Nguyen et al. to handle unbalanced distribution of shadow labels. However, this manual parameter is sensitive to different datasets or situations.

To illustrate the motivation of our method, firstly we revisit several general basic networks for segmentation. The experiments on both CNNs and FCNs for shadows detection are shown in Table 1. We unexpectedly found that shallow CNN with 8 layers produce better results than deeper network with $10-18$ layers. We also compared FCNs with different layer depths and found a similar result. This observation motivates us to consider the basic texture nature of shadows because CNN's lower layers mostly capture image's basic texture. An explicit learning for texture module is reasonably beneficial for shadow detection.

Basic Networks	Feature Layer Depth	BER
FCN-conv5, VGG	16	15.5
FCN-conv4, VGG	13	15.2
FCN-conv3, VGG	10	14.3
CNN, VGG	16	15.3
CNN	8	12.8

Table 1: Experiments on CNNs and FCNs on UCF shadow datasets. FCN experiments are different version of FCN using feature maps from different layers in VGG. CNNs are using patch-based classification task for shadow detection. BER is Balanced Error Rate, smaller is better.

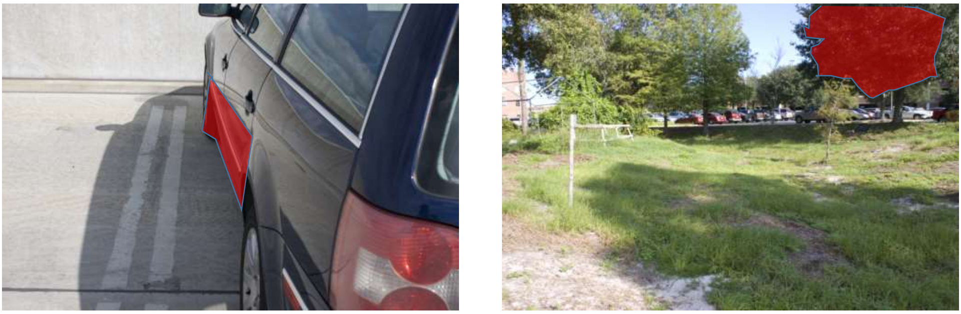

Figure 1: Shadow Texture Learning Challenge. Red masks are non-shadow areas: left is dark area but is not shadow, right is tree area that have similar texture with grass shadow.

One of the challenges for robust shadow texture learning is the various non-shadow textures that are hard to distinguish from shadows. As shown in Figure 1, some non-shadow areas have darker illumination like shadows and some others may have similar textures with shadow areas. Inspired by hard negative mining in the object detection task, we endeavour to find out these texture patches that are hard to distinguish in each image so that we can learn a robust classifier between shadows and nonshadows. In this paper, a novel *hard negative patch mining* module is proposed for this purpose. Different from hard negative method in the object detection, shadow detection is actually a segmentation task which doesn't have an explicit label (positive/negative). It only has labels for each pixel, like shadow and non-shadow. To overcome this obstacle, we propose to use the gradients from back-propagation to estimate the difficulty level for each patch. Our hard negative mining module for segmentation is proven effective for generating reasonable patches through experiments and visualization.

The final architecture is composed of texture learning and semantic learning networks. The texture learning network follows our hard negative patch mining module. The semantic learning network is a structured fully convolutional network. We make up an integrated end-to-end trainable network by utilizing two techniques - patch extraction and stitching backward. Ablation experiments have shown the power of our texture learning module and its improvement for general networks.

In summary, this work mainly contains the three contributions: 1. A novel hard negative patch mining module is proposed for robust texture learning in shadow detection task; 2. An end-to-end trainable network is proposed to integrate texture feature with semantic features; 3. For fair comparison, ablation experiments are carried out and our proposed model achieves a state-of-art performance on most of benchmarks.

2 Related Work

Early works on shadow detection mainly focused on the physical property or illumination feature of shadows or texture retrieval[2]. Finlayson et al. [3] attempted to classify shadow edges based on an illuminant-invariant image. They compared the original shadow image with an intrinsic image and then extracted shadows from the differences. Instead of using intrinsic images, Zhu et al. [15] proposed a statistical learning approach to learn features in detecting shadows in a single image. They utilized intensity, texture nature of shadow regions, and trained a CRF model for labeling shadow pixels. In addition to individual region features, Guo et al. [5] considered pairwise illumination features between regions. They use a region-based graph to represent the original image, whereas a graph-cut is employed to generate labels.

Recent shadow detection methods are significantly advanced by deep learning models. Khan et al. [7] proposed a method to automatically learn shadow features using a seven-layer network architecture. Instead of using superpixel classification, they focused on shadow edges and treated the structured CNN as an edge classifier. Beside using a CNN for edge detection, Shen et al. [13] formalized the problem of pixel labeling as global optimization and recovered shadow regions based on structured label information of edges.

FCN was originally proposed for semantic segmentation. Long et al. [9] replaced full-connected layers by convolutional layers, which transform FCN output into a prediction map instead of one classification label. Vicente et al. [14] trained a FCN as an image-level shadow prior for shadow detection. They used outputs of FCN together with RGB images to train a CNN as patch classifier. Because they divide the stack structure as independent parts, their training process is split. In our work, we utilize the patch mining and stitching methods to propose an end-to-end structure for semantic and texture features.

Vu Nguyen et al. [11] extended conditional GAN to detect shadow mask. They introduced an additional sensitivity parameter to the generator to parameterize the loss of the trained detector. The main insight behind scGAN is that they tried to control the sensitivity of generator which outputs a binary mask. However, most of these deep methods didn't consider the texture feature of shadows, which essentially makes shadows different from other pixel-to-pixel tasks.

3 Proposed Model

We present our overall network in Figure 2. Our overall model is composed by one deep semantic learning network and a texture learning network. The deep semantic network is a fully convolutional network with VGG backbone. The texture learning network takes patches from patch mining module and extract feature maps from deep semanic network. A hard negative patch mining module is proposed to select hard patch candidates for training of each epoch. Outputs of texture and semantic network will finally be integrated into a custom CRF to generate overall predictions.

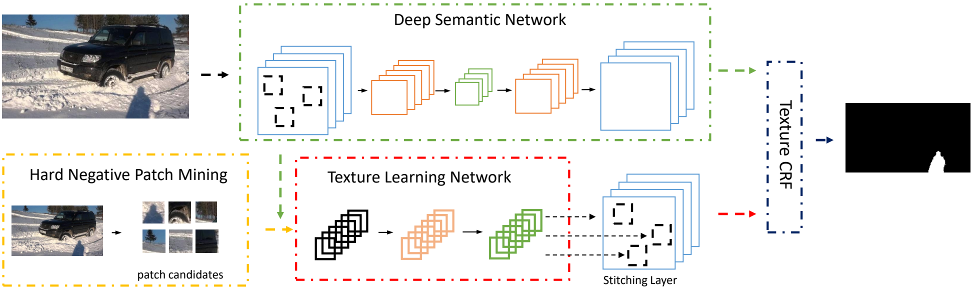

Figure 2: Overall Shadow Detection Network. Our overall model is composed by one deep semantic learning network and a texture learning network. The deep semantic network is a fully convolutional network with VGG backbone. The texture learning network takes patches from patch mining module and extract feature maps from deep semanic network. A hard negative patch mining module is proposed to select hard patach candidates for training of each epoch. Outputs of texture and semantic network will finally be integrated into a custom CRF to generate overall predictions.

3.1 Hard Negative Patch Mining

The hard negative patch mining is proposed for consciously selecting patch candidates that may be hard to distinguish. In the object detection task, one of abilities of hard negative mining method is distribution balance between positive and negative samples. As a statistic clues in UCF shadow dataset, the non-shadow label owns pixels five times that of the shadow label, indicating that shadows are minor part of nature images. Therefore, our mining methods also need to balance patches with different ratio of shadow/non-shadow pixels to achieve a reasonable label amount.

Two key ideas are behind our design of patch mining algorithm for shadow detection. First, the patches should be adaptive for different epoch of training. This one is different from hard negative methods in the object detection. We suppose through the detectors become better and better while training, our method should generate harder and harder patches for it so that we can achieve a more robust detector finally. Second, larger gradients in back-propagation may refer to harder patches for current model. This idea is motivated by the gradients of back-propagation. Large gradient refers to bad predictions in current epoch and bad predictions are probably hard patches for further training.

Algorithm 1 Hard Negative Patch Mining

```
 1: $T$ is the training patches for each epoch. We divide $T$ into $T_r$, $T_s$, $T_p$ for random, shadow
    and candidate patches.
 2: $i \leftarrow 0$
 3: $T_r,\ T_p \leftarrow$ Random patches.
 4: $T_s \leftarrow$ Shadow patches.
 5: **while** $i < num\_epoch$ **do**
 6:     $T_c = T_r + T_p$
 7:     get the backpropagation difference $D_c$ in specific layer for $T_c$
 8:     $T_p \leftarrow$ top $n$ pathchs in $D_c$ by $max(abs(D_c))$
 9:     $T_r,\ T_s \leftarrow$ new sampling patches as line 5.6.
10:     $T = T_s + T_p$
11:     use $T$ for training next epoch.
12:     $i \leftarrow i + 1$
13: **end while**
```

Based on these assumptions, we develop a hard negative patch mining framework to generate texture patches for a robust texture learning. Pseudocode is shown in Algorithm 1.

The main process of our patch mining framework is presented in line $7-10$. After initialization, we collect random patches for each epoch of training and use them to update our hard patch candidates. We treat the last candidate patches together with random patches as current candidates. Shadow patches are sampled every epoch for a balanced distribution betwee shadows and nonshadows. These candidates will then be sorted according to their maximum difference value during back-propagation. Opting to use $max(abs(D_c))$ is mathematical and empirical. Considering the softmax loss function, if we use it and its derivative for patch mining, where z_j is the output of semantic network (to be illustrated in next section), and y_j is the label for jth pixel.

$$L = -\sum_j y_j \log(p_j), \quad p_j = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{1}$$

$$L'_j = \frac{\partial L}{\partial z_j} = p_j - y_j \tag{2}$$

The sorting value $max(abs(D_c))$ is the absolute value of L'_j, which means that we aim to select poor predictions that may include large loss function derivatives.

$$abs(D_c) = |L'_j| = \frac{\partial L}{\partial z_j} = |p_j - y_j|. \tag{3}$$

Given that each patch may contain amounts of outputs that respond to their derivatives, we have experimented *maximum*, *maximum-minimum*, and *average* over these derivatives to estimate each patch. Results show that *maximum* derivatives can better indicate the difficulty level of patches.

3.2 Introduce Texture into Semantic

For current semantic segmentation features, fully convolution networks (FCN) have become a standard model. We introduce two custom layers to incorporate patch mining into FCNs. They are *patch proposal layer* and *stitching layer*.

Given patches coordinate, patch proposal layer generates patch features from FCN feature maps. Its operation is similar to ROI pooling in faster RCNN [12] but not exactly the same. In ROI pooling, the ROIs are collected from RPN or selective search. While in our patch proposal layer, patch ROIs for each epoch are dependent on previous epoch which means $roi_t = f(roi_{t-1})$, where roi_t refers to ROI for particular image in epoch t. To implement this, patch proposal layer needs to restore each image's ROI in previous epoch. The stitching layer is proposed for stitching each patch backward into its corresponding locations to regenerate the feature maps of overall image.

In training, we employ an two-part softmax cross entropy loss function as:

$$L = L_{semantic} + L_{texture} \tag{4}$$

$$L_{texture} = -\sum_{j\ in\ patchs} y_j \log(softmax(P_d(x_j) + P_s(x_j))). \tag{5}$$

$$L_{semantic} = -\sum_{j\ in\ images} y_j \log(softmax(P_d(x_j))), \tag{6}$$

The semantic part accumulates loss of pixels in the whole image, whereas the texture part only involves pixels in selected patches.

In inference, we introduce a custom conditional random field to produce finer prediction results. Basic CRF model can be represented by the following:

$$E(x) = \sum_i \psi_i^U(x_i) + \sum_{i,j} \psi_{ij}^P(x_i, x_j), \tag{7}$$

where $\psi_i^U(x_i)$ and $\psi_{ij}^P(x_i, x_j)$ represent usual unary and pairwise potentials, respectively. In separated CRF model [1], unary potential corresponds to deep learning prediction, and pairwise is RGB color feature. While in our model, texture feature is treated as pairwise similarity between each pair of patches.

we define $\psi_i^U(x_i)$ and $\psi_{ij}^P(x_i, x_j)$ as follows,

$$\psi_i^U(x_i) = P_d(x_i) + P_s(x_i) \tag{8}$$

$$\psi_{ij}^P(x_i, x_j) = w_s Dis(P_s(x_i), P_s(x_j)) + w_r Dis(P_r(x_i), P_r(x_j)). \tag{9}$$

$P_d(x_i)$ and $P_s(x_i)$ represent deep and shallow network predictions, respectively. $P_r(x_i)$ refers to the RGB channel of original images. In Equation 8, unary potential comprises $P_d(x_i)$ and $P_s(x_i)$. We do not visually present weights of these elements as we use an independent scale-learning layer to learn integration weights dynamically. For pairwise potential, we involve shallow texture features as Equation 9. $Dis(x, y)$ refers to the Euclidean distance in our experiments. w_r and w_s represent different weights of our texture feature and RGB feature, respectively.

4 Experiments

4.1 Datasets and Evaluation

We evaluate our method on a standard benchmark UCF [15] shadow dataset (which contains 355 shadow images with their label masks) and a newly collected large SBU dataset [14] (4085 images for the train and 638 for the test). UCF dataset is the most widely used among benchmarks. Thus, we can fairly compare our results with those of previous works. The train/test split follows [5], in which 111 images are used for training, and 110 images are utilized for testing. SBU dataset is newly collected, but its training labels are recovered by algorithm instead of by manual annotation. For fair comparison, we use balanced error rate (BER) as evaluation. We also present shadow/non-shadow to fully compare with other methods.

4.2 General Results

We first present the general results of our texture learning method on UCF and SBU datasets.

Models	Shadow / Non	BER
Paired SVM[5]	26.7 / 6.3	16.5
Stack CNN[14]	10.4 / 12.8	11.6
Semantic (baseline)	17.3 / 9.9	13.4
Semantic+Texture	14.1 / 5.1	**9.6**

Table 2: Our method and state-of-art methods on UCF dataset. Shadow/Non respectively correspond to the error rates of shadow label and non-shadow label.

Models	Shadow / Non	BER
Stack CNN[14]	9.6 / 12.5	11.0
cGAN	20.5 / 6.9	13.6
scGAN[11]	7.8 / 10.4	9.1
Semantic (baseline)	14.3 / 11.7	13.0
Semantic+Texture	9.9 / 6.9	**8.4**

Table 3: Our method and state-of-art methods on SBU dataset.

For fair comparison with previous works, especially that of [14], our baseline model is using FCN-8s as the semantic networks. As shown in the tables, individual semantic segmentation, which is our baseline model, has been better than Paired SVM which is a not deep model but performs worse than most other deep networks.

After involving our texture learning network into semantic networks, the final integrated model achieves a state-of-art accuracy and gets about 36% reduction on balance error rate. Comparing with other state-of-art deep learning models, our method greatly outperforms Stack-CNN which employed a similar basic model with us. One may find that there's a slight accuracy decrease on shadow label, we think it is caused by the mining module's attempt to finding the hardest patches regardless of its labels. It makes our model achieve a much better overall prediction results no matter it is shadow or nonshadow. By Comparing with scGAN which utilized GAN as base structure, our semantic with texture model get a slight improvement while our proposed texture network is a lightweight submodule. We employ a shallow network with only 7 layers to perform basic feature learning which makes it a very cheap module to be added into any other detection tasks. The lightweight submodule also need much less training and inference cost comparing with GAN based models.

4.3 Effectiveness of Hard Mining

We conduct studies on different patch proposal methods to illustrate the effectiveness of our method. Table 4 presents the experimental results on UCF dataset. All trainings share the same initial model and learning rate.

Models	BER
Convnets+CRF [7]	17.7
Stack CNN [14]	11.6
Semantic (Baseline)	13.4
Random Patches	15.3
Shadow Edge Patches	17.8
Canny Edge Patches	15.1
Random + Shadow Edge	11.3
Random + Canny Edge + Shadow Edge	11.1
Hard Negative Patches	**9.6**

Table 4: Different patch proposal experiments on UCF dataset. The + indicates combinations of different methods. *Random Proposal*, *Shadow Edge Patches*, *Canny Edge Patches* respectively refer to selecting patches randomly, selecting patches centered on shadow edges and canny edges patches.

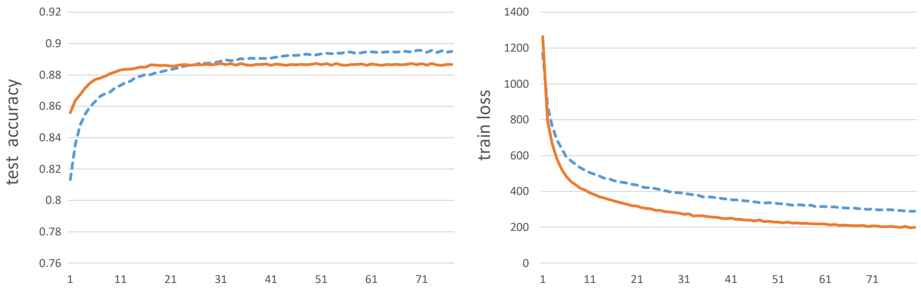

Figure 3: Training visualization between semantic model(dashed line) and model with patch mining(solid line). Left: curve of test accuracy through training. Right: loss of training epochs. As illustration, model with patch mining owns larger training loss but better test prediction accuracy.

We compare our proposal framework with other intuitive or mostly used patch proposal methods. As shown in Table 4, our model achieves the best final detection results compared with any other patch proposal methods. During experiments of patch proposal methods, we discover that our hard negative framework can also efficiently avoid overfitting. Figure 3 shows that hard negative patch mining method exhibits a slow train convergence in training data but present an improved final prediction accuracy in test data, which actually indicates that it can effectively avoid overfitting. This feature is essential and intrinsic in our algorithm, because our proposed framework can always collect harder patches while training.

To better understand the patch mining method, we visualize several images with their training patches for every epoch in Figure 4. As for the last column, we can observe that majority of patches are in shadow edges given that these locations cause difficulty in segmentation. The car image in the last row owns patches from two parts: one in car and one in shadow. In this sample, final patches are not shadow edges but two dark portions that, as we suppose, are hard to distinguish.

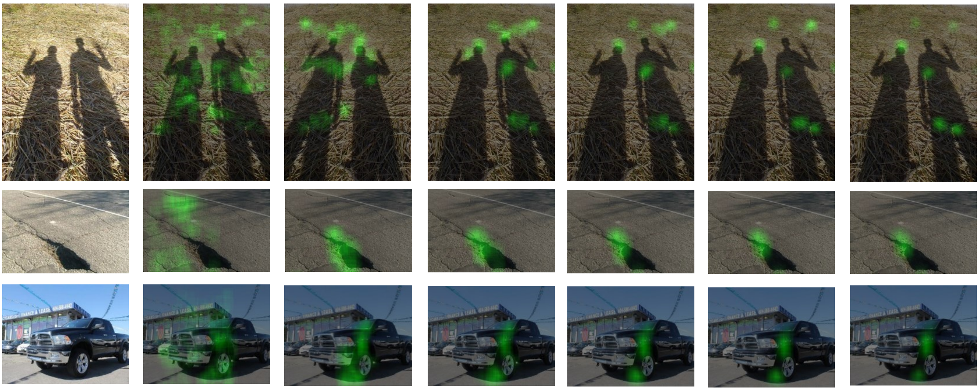

Figure 4: Visualization of Patch Pool Selecting Method. Darker color represents that more patches are proposed in those places. From left to right are original images, proposed patches in first epoch, proposed patches in 10th, 20th, 30th, 40th, 50th epochs.

5 Conclusion and Future work

In conclusion, we propose a novel hard negative patch mining module for robust texture learning in the shadow detection task. Full ablation experiments have been conducted to confirm that our texture learning module can greatly improve the performance of common semantic networks. Our final model also achieves superior final prediction accuracy both in small- and large-scale datasets (UCF and SBU). Comparing with baseline, our patch mining module can give a about 30% performance improvement in error rates. A patching and stitiching framework is proposed to jointly train texture network together with FCNs. It makes our module a lightweight scaffold for other detection tasks, like common semantic segmentation or specific texture detections.

In this paper, we employed VGG16 as our backbone feature learning network for a fair comparison with other methods. In future works, we assume that better basic feature learners, such as ResNet[6], may improve final shadow detection accuracy. The proposed hard negative patch learning may be benificial for other segmentation tasks.

6 Acknowledgements

This work was supported by the NSFC(61672307, 61562063). Bin Wang is the corresponding author. He was partially supported by the National Science Founation of China under Grant 61772301.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[2] Yongsheng Dong, Dacheng Tao, Xuelong Li, Jinwen Ma, and Jiexin Pu. Texture classification and retrieval using shearlets and linear regression. *IEEE Trans Cybern*, 45 (3):358–369, 2015.

[3] Graham D Finlayson, Steven D Hordley, Cheng Lu, and Mark S Drew. On the removal of shadows from images. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):59–68, 2006.

[4] Graham D Finlayson, Mark S Drew, and Cheng Lu. Entropy minimization for shadow removal. *International Journal of Computer Vision*, 85(1):35–57, 2009.

[5] Ruiqi Guo, Qieyun Dai, and Derek Hoiem. Paired regions for shadow detection and removal. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2956–2967, 2013.

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

[7] Salman Hameed Khan, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri. Automatic feature learning for robust shadow detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1939–1946. IEEE, 2014.

[8] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 61(1):1–11, 1971.

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[10] Bruce A Maxwell, Richard M Friedhoff, and Casey A Smith. A bi-illuminant dichromatic reflection model for understanding images. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[11] Vu Nguyen, Tomas F Yago Vicente, Maozheng Zhao, Minh Hoai, and Dimitris Samaras. Shadow detection with conditional generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4520–4528. IEEE, 2017.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[13] Li Shen, Teck Wee Chua, and Karianto Leman. Shadow optimization from structured deep edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2067–2074, 2015.

[14] Tomás F Yago Vicente, Le Hou, Chen-Ping Yu, Minh Hoai, and Dimitris Samaras. Large-scale training of shadow detectors with noisily-annotated shadow examples. In *European Conference on Computer Vision*, pages 816–832. Springer, 2016.

[15] Jiejie Zhu, Kegan GG Samuel, Syed Z Masood, and Marshall F Tappen. Learning to recognize shadows in monochromatic natural images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 223–230. IEEE, 2010.