A FAST AND EFFICIENT NETWORK FOR SINGLE IMAGE SHADOW DETECTION

Leiping Jie[1,2], *Hui Zhang*[2,⋆]

[1]Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, HongKong
[2]Division of Science and Technology, United International College, BNU-HKBU, Zhuhai, P.R. China

cslpjie@comp.hkbu.edu.hk, amyzhang@uic.edu.cn

ABSTRACT

Shadows in images can degrade the performance of many applications. In this paper, we propose a novel multi-level feature-aware network, called **TransShadow**, which uses Transformer to capture both local and global context from a single image for shadow detection. Specifically, we design a multi-level feature-aware module, where multi-level features are selected and processed by the Transformer to distinguish shadowed and non-shadowed regions. To further utilize the remaining feature levels, progressive upsampling with skip connections is proposed to fuse more information for shadow detection. Experimental results show that our approach achieves comparative performance as the state-of-the-art method on benchmark datasets SBU and ISTD with the smallest model size and fastest inference speed. More importantly, our model shows the best generalization performance on the benchmark dataset UCF.

Index Terms— Shadow detection, multi-level feature, transformer, low-level vision

1. INTRODUCTION

Shadows can be seen everywhere in our daily life. They refer to areas where light from a light source is blocked by an opaque object. In many computer vision tasks, such as object tracking, instance segmentation and object detection, shadows are one of the main factors that degrade performance. However, in real-world scenes, different shadows will appear in different areas, with various shapes and intensities. This makes shadow detection a challenging problem.

To tackle this problem, many works have been proposed. Early attempts tried to detect shadows with physical models based on color and lighting [1, 2], or traditional machine learning algorithm with hand-crafted features [3, 4]. The disadvantages of these approaches were the low performance and low robustness. In contrast, recent approaches mostly leveraged deep neural network (DNN) models and achieved superior performance on benchmark datasets [5, 3, 6]. Most of these leading methods built U-Net-like architecture [7].

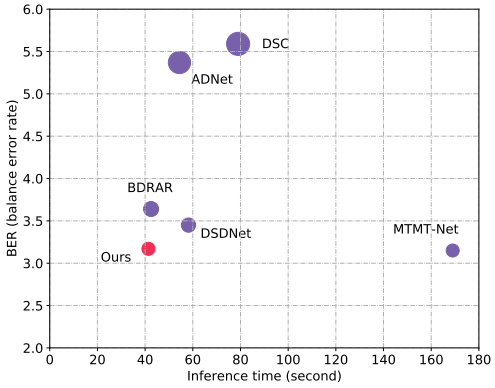

Fig. 1. Average BER and number of parameters with input image size 416x416 on the SBU dataset. The size of the circles represents the BER value of the corresponding approach. Our proposed method achieves the best performance with the fewest parameters, which also shows the fastest inference speed.

Concretely, they first adopted a pretrained backbone, *e.g.* ResNeXt [8], ResNet [9], as feature extractor, and acquired multi-level features at different resolutions. Then, different strategies [10, 11, 12, 13] were developed to fuse the extracted features to predict the shadow map.

As illustrated in [12, 11], local context is not sufficient to distinguish shadow and non-shadow regions. Consequently, it is crucial to take global context into consideration for shadow detection. Nevertheless, state-of-the-art methods only attempt to capture the global context level by level [11, 12], rather than processing multi-level features simultaneously. To overcome this limitation, we design a multi-level feature-aware method to capture more representative global context. In particular, selected multi-level features are aligned to have the same resolution as the largest resolution of the selected features. Then, the aligned features are concatenated along the channel dimension. Next, the concatenated features are flattened and are added with learnable positional embeddings, which will be fed into the Transformer encoder. Since the self-attention mechanism in Transformer is natural for its awareness of global context, our proposed multi-level

⋆ Corresponding author

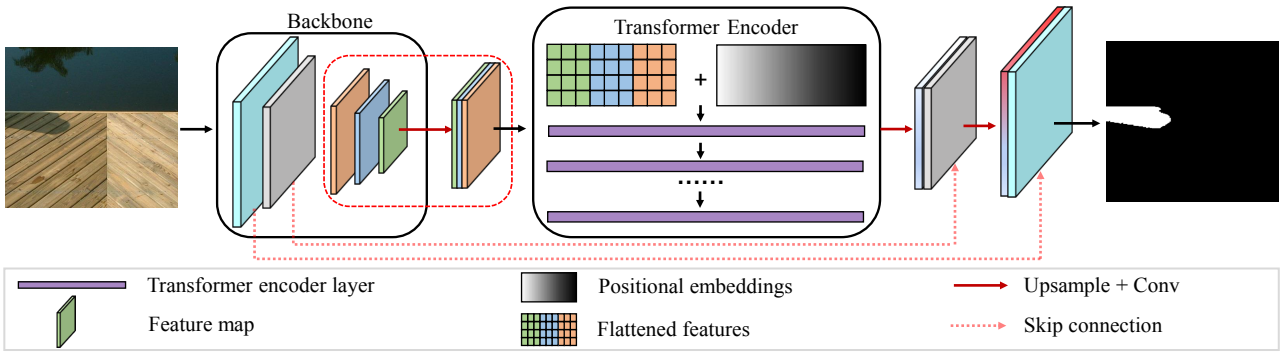

Fig. 2. Illustration of proposed approach. Our model is an encoder-decoder network where the encoder extracts multi-level features and the decoder predicts the shadow map.

feature-aware method is able to be aware of useful global context features among multi-level feature maps. Unlike previous work that split the input images into patches as the input of the Transformer [14], our network captures pixel-to-pixel attention, which is beneficial for dense prediction tasks.

The main contributions of this paper are: (1) We introduce **TransShadow**, a fast, compact and efficient shadow detection framework. (2) We propose a novel multi-level feature-aware module capable of exploring the global context between multi-level features. (3) Our experimental results demonstrate that our model achieves comparative performance as current best approach MTMT-Net [15] with only quarter number of model parameters on two benchmark datasets SBU [5] and ISTD [6]. Meanwhile, our model shows the best performance in terms of generalization on benchmark dataset UCF [3].

2. METHOD

As shown in Fig. 2, our network is composed of the backbone for multi-level feature extraction, the multi-level feature-aware module and the progressive shadow map predictor. We first employ the pretrained EfficientNet B1 [16] to construct a 5-level feature pyramid. Specifically, given an input RGB image $\mathcal{I} \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times C}$, where $\mathcal{H}$, $\mathcal{W}$ and C are height, width and number of channels respectively, the encoder extracts feature maps $\{\mathcal{F}_i\}_{i=1}^{L}$ where $\mathcal{F}_i \in \mathbb{R}^{\mathcal{H}_i\times\mathcal{W}_i\times C_i}$, L is the number of levels, $\mathcal{H}_i = \frac{\mathcal{H}}{2^i}$ and $\mathcal{W}_i = \frac{\mathcal{W}}{2^i}$. From high-level to low-level, the corresponding feature map encodes different kinds of features which will be used in different stage of the decoder.

Compared to previous approaches, we consider multi-level features simultaneously. As a trade-off between speed and model size, we select the highest three levels rather than whole of them. The selected features are fed into our proposed multi-level feature-aware module. After that, progressively upsampling with skip connections will be performed to generate the predicted shadow map.

2.1. Multi-level Feature-aware Module

With extracted n-level features, we select m ($m \leq n$) highest-level features. Due to these features are of different resolutions, we align them to have the same resolution. Specifically, an upsampling operator with two consecutive convolutional layers followed by BatchNorm and Leaky ReLU is applied to the smaller feature maps. To reduce the number of parameters and avoid overfitting, the number of channels is also aligned. After alignment, the features are concatenated along the channel dimension. This procedure can be formulated as follows:

$$\mathcal{F}_c = Conv(\mathcal{F}_{n-m})©UpConv(\mathcal{F}_{n-m+1})©...©UpConv(\mathcal{F}_{n-1}), \tag{1}$$

where ©, $Conv$, $UpConv$ refers to concatenation operator, a convolutional layer with 1×1 kernel and upsampling with two 3×3 convolutional layers respectively, and the aligned feature $\mathcal{F}_c \in \mathbb{R}^{mC_{n-m}\times\mathcal{H}_{n-m}\mathcal{W}_{n-m}}$.

Since the transformer encoder takes sequences as input, we flatten and permute $\mathcal{F}_c$ to $\mathcal{F}_r$ whose shape is $\mathcal{H}_{n-m}\mathcal{W}_{n-m}\times mC_{n-m}$, in which $\mathcal{H}_{n-m}\times\mathcal{W}_{n-m}$ and $m\times C_{n-m}$ can be interpreted as *sequence length* and the *embedding length* respectively. Furthermore, since the spatial information is lost in $\mathcal{F}_r$, we add the permuted features $\mathcal{F}_r$ with the learnable position embeddings $\mathcal{E}_{pos}$ as follows:

$$\mathcal{F}_t = \mathcal{F}_r + \mathcal{E}_{pos}, \tag{2}$$

where $\mathcal{E}_{pos} \in \mathbb{R}^{\mathcal{H}_{n-m}\mathcal{W}_{n-m}\times mC_{n-m}}$ and $\mathcal{F}_t$ is the input to the first Transformer encoder layer.

As shown in Fig. 3, each Transformer encoder layer consists of Layer Normalization (*LN*) [17], Multihead Self-Attention (*MSA*) and Multi-Layer Perceptron (*MLP*). Our Transformer encoder consists of $\mathcal{N}_d$ layers. For any layer $i \in [0, \mathcal{N}_d - 1]$, assuming the input feature is T_i (T_0=$\mathcal{F}_t$), we can get the output feature T_{i+1} as follows:

$$\begin{aligned} T_i^{'} &= MSA(LN(T_i)) + T_i, \\ T_{i+1} &= MLP(LN(T_i^{'})) + T_i^{'}. \end{aligned} \tag{3}$$

Then, the output of our Transformer encoder is reshaped to have resolution $(C_{n-m}, \mathcal{H}_{n-m}, \mathcal{W}_{n-m})$.

Due to global attention in Transformer, our module is able to perceive contextual information from multi-level features, which helps to capture more distinguishable context features for shadow detection.

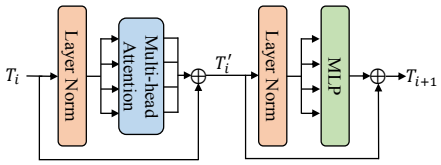

Fig. 3. Illustration of the transformer encoder layer.

2.2. Shadow Map Regression and Loss Function

Progressive Upsampling Our decoder performes progressive upsampling with skip connections. We denote the output of the Transformer as $\mathcal{D}_0$, which is also the input feature for the first stage of our progressive upsampling decoder. Specifically, $\mathcal{D}_i$ is first upsampled 2× by differentiable interpolation. Then, it is concatenated with the skip connection $\mathcal{F}_j$ from the encoder. After that, two consecutive 3×3 convolutional layers with BatchNorm and Leaky ReLU are used to obtain the input features $\mathcal{D}_{i+1}$ for the next stage. To get the predicted shadow image, we perform a 3×3 convolutional layer with output channel one. Note that, for inference, we add an extra Sigmoid layer to constrain the value of the output shadow image in $[0, 1]$. The reason is that we found the Sigmoid operator saturates and kills gradients when training.

Loss Function We adopt focal loss [18] to compensate the unbalanced distribution and focus more on hard samples,

$$\mathcal{L} = \begin{cases} -\alpha(1-y')^{\gamma} log y', & y = 1 \\ -(1-\alpha)y'^{\gamma} log(1-y'), & y = 0 \end{cases} \quad (4)$$

where α, γ, y' and y are the weighting factors for the unbalanced distribution, tunable focusing parameter for the modulating factor $(1-y')^{\gamma}$, the predicted value and the ground truth respectively. Empirically, we set α to $\frac{8}{9}$ and γ to 2.0.

3. EXPERIMENTS

In this section, we compare the proposed method with five state-of-the-art methods: DSC [12], ADNet [19], BDRAR [11], DSDNet [20], MTMT-Net [15]. All experiments are conducted on three popular benchmark datasets: SBU [5], UCF [3] and ISTD [6]. Balance error rate (BER) is adopted as the performance evaluation metric.

3.1. Implementation

Our code is implemented with PyTorch. All training and testing are conducted on a single NVIDIA Tesla V100 GPU.

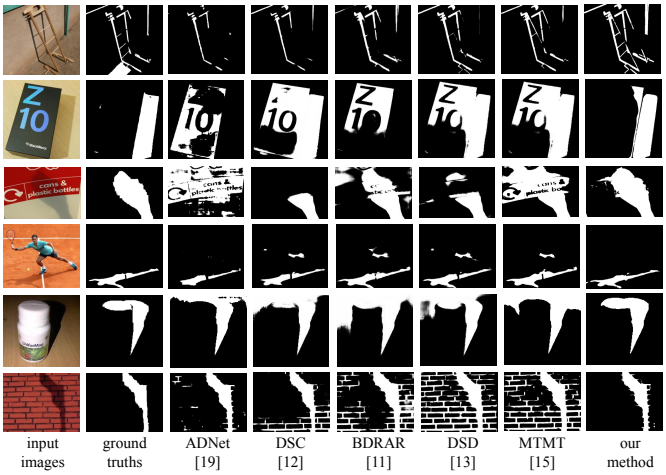

Fig. 4. Qualitative comparison of the predicted shadow maps between our approach and other methods (from the third column to the seventh column) against the ground truths in the second column on SBU [5] benchmark dataset.

Training. 1-cycle policy is adopted for dynamically adjusting the learning rate. In particular, the learning rate is first warm-up from 2×10^{-5} to 6×10^{-5} linearly for the first 10 epochs followed by cosine annealing to 6×10^{-8}. The model is trained 90 epochs using AdamW with weight decay 10^{-2} and batch size is 8. The input images are resized to 416×416 with data augmentation including horizontal flipping, random cropping and brightness adjustment.

Inference. For testing, the input image is also resized to 416×416 as training and the predicted shadow map is resized back to the original input image size for evaluation. Unlike other methods, we do not perform any post-processing operations, *e.g.* conditional random field (CRF).

3.2. Quantitative and Qualitative Results

In this section, we compare our results with the five aforementioned state-of-the-art methods. For fair comparison, quantitative results of other methods are reported as in their paper, while the qualitative results are either generated using official implementations and pretrained models provided by the authors, or provided directly by the authors.

Quantitative Comparison. As shown in Table 1, our method achieves comparative results as state-of-the-art method MTMT-Net [15] on SBU [5] and ISTD [6] dataset. However, our model has only 1/4 parameters compared with MTMT-Net [15] and our model is trained on the SBU dataset only while MTMT-Net enlarges the training dataset with unlabeled data. Moreover, when comparing to other methods with parameters of similar scale, *e.g.* DSDNet [20], BDRAR [11], ADNet [19], DSC [12], our model has a better performance in terms of BER. Specifically, our method has 8.41%, 13.2%, 41.2% and 43.5% lower BER scores respectively. This indicates that our model is compact and efficient. On the UCF

Table 1. Quantitative comparison results on SBU [5], UCF [3] and ISTD [6] dataset. The best and the second best results are highlighted in red and blue respectively.

Method	Params (M) Time (s)	BER ↓		
		SBU [5]	UCF [3]	ISTD [6]
scGAN [10]	-/-	9.10	11.50	4.70
ST-CGAN [6]	-/-	8.14	11.23	3.85
DSC [12]	79/0.091	5.59	10.54	3.42
ADNet [19]	54.4/0.031	5.37	9.25	-
BDRAR [11]	42.5/0.026	3.64	7.81	2.69
DSDNet [20]	58.2/0.023	3.45	7.59	2.17
MTMT-Net [15]	169/0.041	3.15	7.47	1.72
Ours	41.4/0.017	3.17	6.95	1.73

[3] benckmark, our model outperforms MTMT-Net, DSD-Net, BDRAR, ADNet, DSC, scGAN, ST-CGAN by 6.83%, 8.30%, 10.88%, 24.76%, 33.97%, 39.48%, 46.46% respectively. Note that we evaluate on the UCF dataset using model trained on the SBU dataset without any fine tuning, which convinces the generalization performance of our model.

Qualitative Comparison. Fig. 1 shows our predicted shadow maps compared with other approaches. Due to our proposed multi-level feature-aware module, our method is able to capture more information for distinguishing whether a pixel is a shadow pixel or not. As shown in the second row, the shadow is directly casted by the black box. Without the understanding of global context, it is hard to discriminate between the two objects. In the last row, the shadow is projected on the wall. Other methods are unable to judge the texture on the wall except our method.

Model Size and Speed As shown in Table 1, our model runs the fastest. The size of our model is smaller by 90.82%, 31.4%, 2.66%, 40.58%, 308.21%, when comparing with DSC, ADNet, BDRAR, DSDNet, MTMT-Net respectively. Nevertheless, we still show a better performance than all other approaches except MTMT-Net which is around $4x$ of model parameters to ours.

3.3. Ablation Studies

In this section, we conduct ablation studies for extensive evaluation and analysis of our approach.

Number of Selected Levels We first evaluate the effectiveness of our proposed multi-level feature-aware module. Note that the larger the number of stages, the smaller the resolution. We first design a baseline network with only progressive upsampling and skip connections. As we can select different numbers of multi-level features as the input of our multi-level feature-aware module, we construct three variants: 1) $5th$-level, 2) $5th$-level and $4th$-level, 3) $5th$-level, $4th$-level and $3th$-level respectively. Here, ith-level represents the ith-level feature extracted by the encoder. We train all these four networks using the SBU training split and evaluate them on the SBU testing spit. As we can see from Table 2, the performance gradually increases as more levels of features are considered simultaneously by the Transformer encoder. Note that we only evaluate up to 3 levels since more levels will lead to high GPU memory consumption.

Table 2. Ablation studies of baseline network and its variants.

$5th$-level	$4th$-level	$3th$-level	Params (M)	BER ↓
×	×	×	36.8	3.68
✓	×	×	39.5	3.36
✓	✓	×	40.2	3.29
✓	✓	✓	41.4	**3.17**

Transfomer Encoder Settings Two important hyperparameters in our Transformer encoder are the number of heads $\mathcal{N}_h$ and the number of depth $\mathcal{N}_d$. Note that the number of depth affects the number of model parameters while the number of heads does not. We use different $[\mathcal{N}_h, \mathcal{N}_d]$ combinations, $[4,4]$, $[4,8]$, $[8,8]$, $[8,12]$. As shown in Table 3, $[4,8]$ performs the best while $[8,12]$ is worst. We think the reason why larger model performs worse is because of the scale of our training dataset. As introduced in [5], SBU benchmark has only $4,089$ training images which is relatively small. Another possible reason is that we only perform three simple data augmentation operations.

Table 3. Ablation studies of different settings for the number of heads $\mathcal{N}_h$ and the number of depth $\mathcal{N}_d$.

Number of Heads	Number of Depth	Params (M)	BER ↓
4	4	25.7	3.45
4	8	41.4	**3.17**
8	8	41.4	3.29
8	12	57.2	3.36

4. CONCLUSION

This paper presents a novel, fast and efficient network TransShadow for single-image shadow detection. Our key idea is to consider multi-level features simultaneously instead of one by one. We propose a multi-level feature-aware module to explore more representative global and local context. Experimental results demonstrate that our model achieves comparable performance as the current state-of-the-art on two benchmark datasets, SBU and ISTD, using only a quarter of the number of model parameters. Meanwhile, our model shows the best performance in terms of generalization on the benchmark dataset UCF.

5. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (62076029), the Natural Science Foundation of Guangdong Province (2017A030313362) and two internal funds of the United International College (R202012, R201802).

6. REFERENCES

[1] 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*, vol. 28, no. 1, pp. 59–68, 2006.

[2] Graham D Finlayson, Mark S Drew, and Cheng Lu, "Entropy minimization for shadow removal," *International Journal of Computer Vision*, vol. 85, no. 1, pp. 35–57, 2009.

[3] Jiejie Zhu, Kegan GG Samuel, Syed Z Masood, and Marshall F Tappen, "Learning to recognize shadows in monochromatic natural images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, p. 223–230.

[4] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Detecting ground shadows in outdoor consumer photographs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010, pp. 322–335.

[5] Tomas 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, p. 816–832.

[6] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, p. 1788–1797.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[8] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 1492–1500.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2016, pp. 770–778.

[10] Vu Nguyen, Tomas F Yago Vicente, Maozheng Zhao, Minh Hoai, and Dimitris Samaras, "Shadow detection with conditional generative adversarial networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017, p. 4510–4518.

[11] Lei Zhu, Zijun Deng, Xiaowei Hu, Chi-Wing Fu, Xuemiao Xu, Jing Qin, and Pheng-Ann Heng, "Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 121–136.

[12] Hu Xiaowei, Lei Zhu, Chi-Wing Fu, Jing Qin, and Pheng-Ann Heng, "Direction-aware spatial context features for shadow detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7454–7462.

[13] X. Hu, C. W. Fu, L. Zhu, J. Qin, and P. A. Heng, "Direction-aware spatial context features for shadow detection and removal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 11, pp. 2795–2808, 2020.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020.

[15] Zhihao Chen, Lei Zhu, Liang Wan, Song Wang, Wei Feng, and Pheng-Ann Heng, "A multi-task mean teacher for semi-supervised shadow detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5611–5620.

[16] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[17] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, "Layer normalization," 2016.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[19] Hieu Le, Tomas F. Yago Vicente, Vu Nguyen, Minh Hoai, and Dimitris Samaras, "A+D Net: Training a shadow detector with adversarial shadow attenuation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 662–678.

[20] Quanlong Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau, "Distraction-aware shadow detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, p. 5167–5176.