Distraction-aware Shadow Detection

Quanlong Zheng Xiaotian Qiao Ying Cao Rynson W.H. Lau
Department of Computer Science, City University of Hong Kong

Abstract

Shadow detection is an important and challenging task for scene understanding. Despite promising results from recent deep learning based methods. Existing works still struggle with ambiguous cases where the visual appearances of shadow and non-shadow regions are similar (referred to as distraction in our context). In this paper, we propose a Distraction-aware Shadow Detection Network (DSDNet) by explicitly learning and integrating the semantics of visual distraction regions in an end-to-end framework. At the core of our framework is a novel standalone, differentiable Distraction-aware Shadow (DS) module, which allows us to learn distraction-aware, discriminative features for robust shadow detection, by explicitly predicting false positives and false negatives. We conduct extensive experiments on three public shadow detection datasets, SBU, UCF and ISTD, to evaluate our method. Experimental results demonstrate that our model can boost shadow detection performance, by effectively suppressing the detection of false positives and false negatives, achieving state-of-the-art results.

1. Introduction

Shadow is a common illumination phenomenon, caused by the occlusion of light by some occluders, resulting in color and intensity changes in the local surfaces. Hence, detecting shadows allows us to infer, for example, the light source position [19, 26], scene geometry [16, 25, 15], and camera parameters [37]. On the other hand, shadows can impede many vision tasks, *e.g.,* visual tracking [2], object detection [23], and semantic segmentation [6]. Hence, shadow detection has been long-time studied and is a significant computer vision problem.

Early shadow detection methods are mostly physical models mainly based on color chromaticity or illumination invariant assumptions and use hand-craft features *e.g.,* illumination cues [3, 27, 7], color [4, 30], and others [14, 41]. With the success in applying deep learning on vision tasks, recent data-driven models [17, 32, 28, 24, 42] learn to detect shadows using CNNs. They achieve significant performance improvements over the traditional ones. The mainstream methods mainly take two strategies, *i.e.,* enlarging the training data [21] or combining global context information [24, 34, 42].

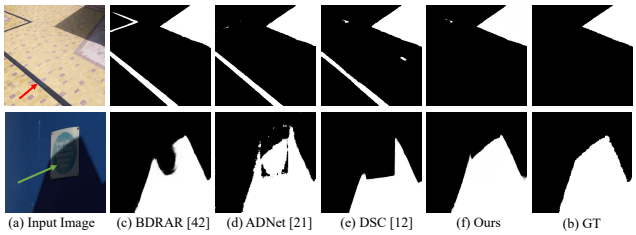

Figure 1. Shadow detection with distraction. Existing methods [42, 21, 12] wrongly detect a non-shadow region in the input image that appears like shadow (pointed to by a red arrow in (a) top row) as shadow ((b-d) top row), and also wrongly consider a shadow region that appears like a non-shadow pattern (pointed to by a green arrow in (a) bottom row) as non-shadow ((b-d) bottom row). Our distraction-aware model can detect shadow regions favorably in both cases. Best view in color.

In natural images, there are often non-shadow regions in the image that appear like shadows (*e.g.,* Figure 1(a) top row) and hence are wrongly detected as shadows (*e.g.,* Figure 1(b-d) top row), and there are also shadow regions that appear like non-shadow patterns (*e.g.,* Figure 1(a) bottom row) and hence are wrongly considered as non-shadows (*e.g.,* Figure 1(b-d) bottom row). In this paper, we refer to these ambiguous regions as *distraction*, and consider two types of distractions: false positive distraction (FPD) - shadow-like non-shadow regions, and false negative distraction (FND) - shadow regions with non-shadow patterns. State-of-the-art methods [42, 21] tend to misdetect these ambiguous regions, partially because their models treat all the cases equally, thereby requiring significant amounts of data to learn to resolve such challenging ambiguities. Hence, limited availability of labeled shadow detection data [32] hinders the performance of these models.

We note that humans can often detect shadows very accurately, without being affected by distractions. Our obser-

Ying Cao is the corresponding author. This work was led by Rynson Lau.

vation is that humans would first recognize the distraction semantics (*e.g.,* shapes, patterns, objects), which are important cues to resolve the ambiguities, and then compare them with confident shadow regions to make the final decision. For example, humans consider the dark region (pointed to by a red arrow in Figure 1(a) top row) as non-shadow since we recognize that region is a part of a texture pattern and is heterogeneous with the shadow at the top right hand corner. On the other hand, we identify the brighter region (pointed to by a green arrow in Figure 1(a) bottom row) as shadow since this part of the logo is homogenous with other part of the shadow in terms of intensity, chromaticity and direction. Thus, we hypothesize that explicitly modeling the semantic features of potential ambiguous regions could help boost shadow detection performance.

In this paper, we propose to detect shadows in a distraction-aware manner. We design a Distraction-aware Shadow (DS) module to learn and integrate distraction explicitly into the shadow detection task. The DS module takes as input the visual features of an input image and fuses them with FND and FPD features to output distraction-aware, discriminative features for shadow detection. It first learns to extract FND and FPD features induced by visual ambiguities in images, and then integrates the distraction features using two different strategies according to the characteristics of the two types of distractions. Specifically, FND often has a variable visual appearance that is very different from general shadows, making it very difficult to be detected. Therefore, we use an attention mechanism to select the features around potential FN regions and add them to the input image features to obtain FN-augmented features, enabling the model to better discriminate FNs. On the other hand, FPD is often locally similar to general shadows, and requires a broader context in order to capture its characteristics. Hence, we apply a Conv block to obtain FP-aware features with a larger receptive field, which are then subtracted from the FN-augmented features to make the model less vulnerable to FPs. The DS module is inserted into a multi-scale framework to extract distraction-aware shadow features at different scales to predict a shadow detection map. Experimental results on three public shadow detection datasets, SBU, UCF and ISTD, show that our method outperforms the prior methods and achieves state-of-the-art results.

The main contributions of this paper are as follows. First, we introduce the concept of distraction to the shadow detection problem, allowing more accurate detection of shadows. Second, we propose a distraction-aware shadow module to integrate the distraction semantics into our end-to-end multi-scale shadow detection framework. We experimentally demonstrate that our model achieves the state-of-the-art shadow detection performance.

2. Related Work

In this section, we focus our review on single image shadow detection methods.

Traditional methods. Early works developed physical models based on illumination invariant assumptions [4, 3]. Such assumptions can only work on high-quality and well-constrained images, while perform badly on complex consumer photographs. Later, more works [14, 20, 41] were proposed for consumer photographs based on various hand-crafted features. Edge and pixel information were first explored. For example, Zhu *et al.* [41] trained a classifier based on texture, gradient and intensity cues. Huang *et al.* [14] trained a shadow detector via feeding the edge features in SVM [9]. Subsequently, instead of separately using the pixel level cues, the region level cues were explored. For example, Guo *et al.* [7] computed illumination features for segmented regions and then built a graph-based classifier using both the individual region information and pairwise relations. Vicente *et al.* [33] trained classifiers for shadow and non-shadow regions, and applied MRF to enhance the performance by utilizing the pairwise region context. All of the above methods are based on hand-crafted features, which are not discriminative enough in complex scenes.

Deep learning based methods. Recently, deep learning based shadow detection methods become very popular, due to the success of deep learning in computer vision tasks [8, 29, 22]. In the beginning, researchers mainly treated the CNN as a powerful feature extractor and made significant performance improvement with the powerful deep features. Khan *et al.* [26] first applied the CNN to shadow detection. They utilized a 7-layer CNN to extract features from superpixels and then feed the features to a CRF model to smooth the detection results. Shen *et al.* [31] first extracted shadow edges via a structured CNN, and then solved shadow recovery as an optimization problem. Later, end-to-end CNN models were proposed due to the emergence of fully convolutional networks (FCN) [22]. For example, Vicente *et al.* [32] presented a semantic-aware stacked CNN model to extract the semantic shadow prior and then refined the output by a patch-based CNN.

More recently, contextual information is being explored. In [24, 34], the generative adversarial networks (GANs) [5] were adopted to capture contextual information, since the adversarial training strategy enables the generator to mimic the distribution of the whole dataset. In particular, Nguyen *et al.* [24] presented a scGAN model, where a sensitivity parameter was introduced to the generator to control the sensitivity of the shadow detector. Wang *et al.* [34] proposed a stacked cGAN model to jointly learn shadow detection and removal. Le *et al.* [21] generated challenging samples by attenuating shadow regions via an adversarial training strategy, and the generated samples together with the original images were used to train the detector. Un-

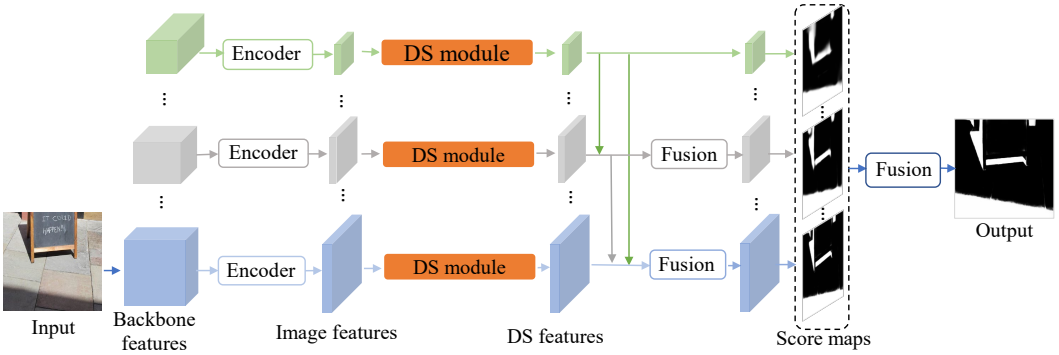

Figure 2. Network Architecture. DSDNet takes as input an image and outputs a shadow map in an end-to-end fashion. First, a backbone is applied to the image to obtain features at different scales. At each scale, the backbone features are sent into an encoder to get the image features. After that, a DS module is applied to generate the DS features. In each scale, we concatenate and fuse its DS features with the up-sampled DS features from smaller scale streams to predict a set of shadow score maps. Finally, we fuse all the shadow score maps to generate a final prediction map.

like the above context acquisition methods, Hu *et al.* [12] proposed to explore spatial context in a directional-aware manner and adopted a RNN-based module to learn spatial context in four directions. Zhu *et al.* [42] proposed a bidirectional recurrent model to combine both global context and local context for shadow detection. Wang *et al.* [36] proposed to aggregate multi-scale context with dense connections. However, these context based methods still suffer on images with complex backgrounds, since they use the context to help minimize the differences between the ground truth and predictions, which tend to satisfy the most common cases while ignoring the challenging cases. In this paper, we take distraction semantics into account and propose a distraction-aware model, which is robust for shadow detection on images with complex backgrounds.

Distraction in computer vision tasks. Distraction cues have been explored in several vision tasks, *e.g.,* semantic segmentation [13], saliency detection [1, 38], and visual tracking [43]. Existing works employ the distraction cues either to filter out the distracting input regions [38] or suppress negative high-level representations [13, 1, 43]. Unlike the above methods, we split shadow distraction into two types, FND and FPD, and design specific architectures to efficiently integrate the two types of distraction semantics. To the best of our knowledge, it is the first work to introduce distraction semantics in shadow detection.

3. Our Approach

Figure 2 shows the architecture of the proposed DSDNet. The network is built on the DSS [11]. As in [42], we choose ResNeXt-101 [39] as our backbone network. DSDNet takes as input an image and predicts a shadow map in an end-to-end fashion. First, the input image is fed into the backbone network to extract backbone features at different scales. Similar to [11, 40], we use the outputs of the last convolutional layer of each stage of the backbone network, *i.e.,* conv1, res2c, res3b3, res4b22 and res5c, as the backbone features. At each scale, an encoder transforms the backbone features to image features. Each DS module will then take as input the image features and produce DS features, which capture the distraction semantics. After that, the DS features are up-sampled via bilinear interpolation, then merged by concatenation from top to down with dense connections, and finally sent to a convolution layer for fusion, like [11]. Specifically, let F_k be the up-sampled features for scale k. The merged features at the current scale can be obtained via: $F_k^m = Conv(Concat(F_k, ..., F_1))$. Finally, all the shadow score maps are fused with a 1x1 convolutional layer followed by a sigmoid activation function to output a soft binary shadow map as the final output.

3.1. Distraction-aware Shadow (DS) Module

As shown in Figure 3, the input to the DS module is the image features, $f_{im} \in R^{H \times W \times 32}$, extracted by any CNNs and the output is the DS features, $f_{ds} \in R^{H \times W \times 32}$. The DS module is designed to explicitly learn semantic features of the potential distraction regions and fuse the distraction features with the input image features to produce distraction-aware features that will be used for shadow detection. It mainly consists of a FN sub-module and FP sub-module, as well as some operations (red lines) to combine different features. The FN sub-module and FP sub-module are discussed in detail below.

FN sub-module. It is designed to learn FN features, $f_{fnd} \in R^{H \times W \times 32}$, and FN-masked features, $f'_{fnd} \in R^{H \times W \times 32}$, which are used to augment the input image features, f_{im}. It first employs a feature extractor on the image features to extract the FN features. To force the FN fea-

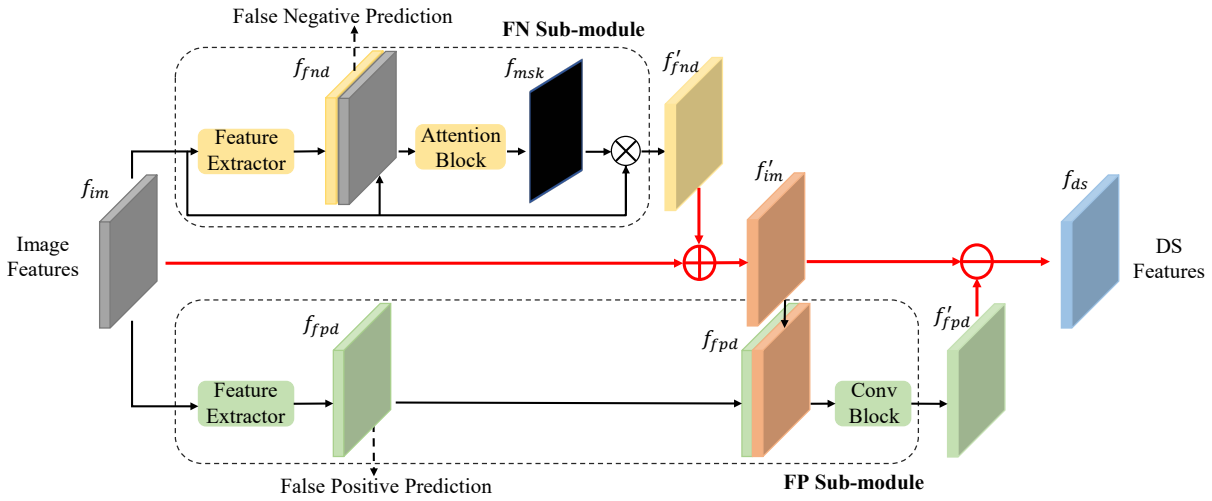

Figure 3. Architecture of the Distraction-aware Shadow module (DS module). Image features (f_{im}) are passed through FN sub-module to produce FN-masked image features ($f^{'}_{fnd}$), which are added with f_{im} to produce FN-augmented features $f^{'}_{im}$. FP sub-module takes as input f_{im} and $f^{'}_{im}$ and outputs FP-aware image features $f^{'}_{fpd}$, which are subtracted from $f^{'}_{im}$ to get distraction-aware features f_{ds}.

tures to capture the semantics necessary to recognize potential FN regions, we use the FN features for FN prediction, by estimating a soft binary map indicating the possible FN locations on the input image. The FN features are then concatenated with the image features and fed into an attention block to produce a soft mask, $f_{msk} \in [0,1]^{H\times W}$. Then, a masked image presentation $f^{'}_{fnd}$ is obtained by multiplying f_{im} with duplicated f_{msk} (along feature channel) element-wise. To enhance the feature activations on FN regions, $f^{'}_{fnd}$ are added to f_{im} to produce FN-augmented image features, $f^{'}_{im} \in R^{H\times W\times 32}$. The attention mechanism is designed to enable the network to quickly focus on and augment the features around possible FN regions. This would help the network better discriminate the FN regions, whose visual appearances are highly variable and differ greatly from general shadows.

FP sub-module. Similar to the FN Sub-module, the FP Sub-module is used to learn the FP features, $f_{fpd} \in R^{H\times W\times 32}$, and FP-aware features, $f^{'}_{fpd} \in R^{H\times W\times 32}$, which are used to further enhance the FN-augmented features. It also adopts a feature extractor with the same architecture as that of the FN Sub-module to extract f_{fpd}. To force the FP features to capture useful semantics of potential FP regions, we predict a soft binary map of false positives in the same way as in the FN Sub-module. We then concatenate f_{fpd} with $f^{'}_{im}$ and feed it into a Conv block to generate the FP-aware image features, $f^{'}_{fpd}$, which capture the characteristics of false positives. Due to the local similarity of FP regions to real shadows, we use a Conv block composed of several convolution layers to capture larger context information, which are useful in discriminating between FP regions and real shadows. Finally, we subtract $f^{'}_{fpd}$ from $f^{'}_{im}$ to eliminate the negative effect of FP features on detection. This would make the network less susceptible to possible FP distraction.

3.2. Deriving Distraction Supervision

To train our network, we need supervision on both FP and FN predictions from our DS module. Unfortunately, annotating false positives and false negatives is both expensive and subjective. We thus propose to acquire the approximate FN and FP ground truth based on the differences between the predictions from existing shadow detection models and their ground truth. As a single model is difficult to generate all representative distractions, we combine the predictions from multiple existing methods for our purpose. In particular, we choose a baseline model and several recent models ([12, 42, 21]) to generate the predictions. For each image, we first compute a difference map between the prediction from each of the models and the corresponding ground truth shadow map. We then combine all the difference maps to form a final difference map, where false positives and false negatives are considered as the ground truth. We have found that such an approach of approximating false positives and false negatives works well with our model, which will be demonstrated in Section 4.5.

3.3. Training

We train our network to jointly optimize the predictions of shadow, FN and FP maps at all scales, and the final shadow map as well as the final FN and FP maps (where the final FN/FP map is obtained by fusing the FN/FP map

predictions of the DS module at different scales) by minimizing the objective:

$$L = \alpha \sum_i L^i_{shadow} + \beta \sum_i L^i_{fn} + \lambda \sum_i L^i_{fp} + \alpha L^F_{shadow} + \beta L^F_{fn} + \lambda L^F_{fp}, \quad (1)$$

where L^i_{shadow}, L^i_{fn} and L^i_{fp} are the losses for the shadow, FN and FP map predictions at i-th scale, respectively. L^F_{shadow}, L^F_{fn} and L^F_{fp} are the losses for the final shadow, FN and FP map predictions, respectively.

We denote the shadow probability of the j-th pixel as p_j and its ground truth as y_j ($y_j = 1$ for shadow pixel, $y_j = 0$ for non-shadow pixel). Our shadow loss for scale i is formulated, *i.e.,* $L^i_{shadow} = l_1 + l_2$. l_1 is a weighed cross entropy loss, with class re-balancing weights to address the fact that there are usually significantly more non-shadow pixels than shadow pixels in an image:

$$l_1 = \sum_j (-\frac{N_n}{N_n + N_p} y_i log(p_j) - \frac{N_p}{N_n + N_p}(1-y_j) log(1-p_j)), \quad (2)$$

where j indexes over all the pixels in the image. N_n and N_p denote the number of FN and FP pixels, respectively. l_2 is a distraction-aware cross entropy loss (DS loss), which gives more penalty to the distraction pixels to force the network to focus more on the regions prone to misdetection.

$$l_2 = \sum_j (-\frac{N_n}{N_n + N_p} y^{fnd}_j y_i log(p_j) - \frac{N_p}{N_n + N_p} y^{fpd}_j (1 - y_j) log(1 - p_j)), \quad (3)$$

where y^{fnd}_j is the ground truth of a FN pixel, and y^{fpd}_j is the ground truth of a FP pixel. For L^i_{fn}, L^i_{fp}, L^F_{fn} and L^F_{fp}, we use the same weighted cross entropy loss as in Eq. 2. L^F_{shadow} is identical to L^i_{shadow}.

4. Experiments

In this section, we first introduce the implementation details (Section 4.1), evaluation datasets and evaluation metrics (Section 4.2). We then compare our results both quantitatively and qualitatively to the existing shadow detection methods (Section 4.3) and salient object detection methods (Section 4.4). We further analyze the effects of incorporating distraction semantics (Sections 4.5), and conduct thorough ablation studies to analyze the proposed model (Section 4.6). Finally, we analyze the effects of various strategies for generating distraction supervision (Section 4.7).

4.1. Implementations Details

Network details. Our model is implemented in PyTorch. As mentioned above, the ResNext-101 is chosen as the backbone network for fair comparison with the latest work [42]. The convolutional layers used in our network, unless otherwise stated, are all followed by a Batch Normalization layer and ReLu activation function. For the encoder in Figure 2, it is composed of 2 convolutional layers, each with 32 kernels of size 3×3. For both FN Sub-module and FP Sub-module, the feature extractor has 2 convolutional layers with 32 kernels of size 3×3. The attention block in the FN Sub-module has one convolutional layer with 64 kernels of size 3×3, followed by a sigmoid activation function. As for the Conv block in the FP Sub-module, it is composed of 1 residual block (where the 3 convolutional layers have 64 filters each, with kernel size = $1 \times 1, 3 \times 3$ and 1×1), followed by another 3 convolutional layers (64 filters in the first layer and 32 filters in the other two) with kernel size = $1 \times 1, 3 \times 3$ and 1×1.

Training and inference details. The ResNext-101 is pre-trained on the ImageNet and the other parameters are randomly initialized. We optimize the parameters by the SGD optimizer with momentum = 0.9, weight decay = 5×10^{-4}, with a batch size of 10. We set the initial learning rate to 5×10^{-3} and decrease it by the polynomial strategy with a power of 0.9. The training data is augmented by random horizontal flipping, and the image is resized to 320×320. We train our model for 5,000 iterations. For the loss weights, we set α to 1, β and λ to 4. For inference, we resize the input images to 320×320 and feed them to our model to predict the outputs. Finally, we apply CRF [18] as post-processing to smooth the outputs.

4.2. Dataset and Evaluation Metrics

Datasets. We use three public datasets, UCF [41], SBU [32] and ISTD [32] to evaluate our method. The UCF dataset contains 245 images and 110 of them are used for evaluation. The SBU dataset is the largest shadow dataset covering general scenes, with 4,089 training images and 638 testing images. ISTD is a recently proposed dataset used for both shadow detection and removal. It contains 1,870 triples of shadow images, shadow maps and shadow-free images, and 540 of them are used for testing. We only use their images and shadow maps for shadow detection evaluation. All the testing images provide pixel-wise annotations. Note that for SBU and UCF, we train our model on the SBU training dataset, and test on both SBU and UCF as [42, 12, 21]. ISTD only contains cast shadow images (where the objects casting the shadows are not visible), which are different from those in SBU. Hence, we retrain all models on the ISTD training dataset and evaluate their performances on the test dataset.

Evaluation Metrics. We adopt the balance error rate (BER) [33] to quantitatively evaluate the results as follows:

$$BER = 1 - 0.5 \times (\frac{N_{tp}}{N_p} + \frac{N_{tn}}{N_n}), \quad (4)$$

	SBU			UCF			ISTD		
methods	BER	Shadow	Non Shad.	BER	Shadow	Non Shad.	BER	Shadow	Non Shad.
DSDNet (Ours)	**3.45**	**3.33**	3.58	**7.59**	9.74	5.44	**2.17**	1.36	2.98
BDRAR [42]	3.64	3.40	3.89	7.81	9.69	5.94	2.69	**0.50**	4.87
ADNet [21]	5.37	4.45	6.30	9.25	8.37	10.14	-	-	-
DSC [12]	5.59	9.76	**1.42**	10.54	18.08	**3.00**	3.42	3.85	3.00
ST-CGAN [34]	8.14	3.75	12.53	11.23	**4.94**	11.23	3.85	2.14	5.55
scGAN [24]	9.04	8.39	9.69	11.52	7.74	15.30	4.70	3.22	6.18
Stacked-CNN [32]	10.80	8.84	12.76	13.0	9.0	17.1	8.60	7.96	9.23
RAS [1]	7.31	12.13	2.48	13.62	23.06	4.18	11.14	19.88	2.41
SRM [35]	6.51	10.52	2.50	12.51	21.41	3.60	7.92	13.97	**1.86**

Table 1. Quantitative shadow detection results. We compare our method with state-of-art shadow detection methods, BDRAR [42], ADNet [21], DSC [12], ST-CGAN [34], scGAN [24] and Stacked-CNN [32], and salient object detection methods, RAS [1] and SRM [35]. A lower value indicates a better performance. The best results are highlighted in bold.

where N_{tp}, N_{tn}, N_n and N_p denote the numbers of true positives, true negatives, shadow pixels and non-shadow pixels, respectively. BER is effective for evaluation the class imbalance results and thus is widely used for shadow evaluation. A lower score indicates a better performance.

4.3. Comparison with Shadow Detection Methods

We compare our method with state-of-the-art shadow detection methods: ADNet [21], BDRAR [42], DSC [12], ST-CGAN [34], scGAN [24] and stackedCNN [32]. For fair comparison, we use the results on SBU and UCF from the authors (except for DSC and BDRAR as they provide their results on a different test split from ours; hence we run their provided models on UCF to get the results). For ISTD, we re-train DSC and BDRAR on the ISTD training dataset with their provided codes. Unfortunately, since we are not able to get the results or codes of the other methods, we adopt the evaluation values reported in ST-CGAN [34]. (For ADNet, since we are not able to get the training code or the evaluation values. We are unable to report any results.) Table 1 shows the quantitative comparison results. It shows that our method has the best BER scores on all three test datasets. Compared to the best-performing existing method, BDRAR, our method has 5.22% and 2.82% lower BER scores on SBU and UCF, respectively. This suggests that our model has a good generalization ability. In addition, our model has a closer BER score on shadow and non-shadow pixels. A possible reason is that as we explicitly consider two types of distractions, it forces the network to uniformly balance its performance between shadow and non-shadow regions.

We further show some visual results to qualitatively compare our method with the existing methods, as shown in Figure 4. From the top three rows, we can see that, in comparison to other methods, our method has better ability to discriminate between true shadows and non-shadows with shadow appearance (potential false positives) . For example, in the first row, DSDNet can accurately detect the shadow regions, while some of the existing methods (*e.g.,* ADNet) tend to mis-detect the black shorts of the tennis player as shadows. In the third row, existing methods (*e.g.,* ADNet and BDRAR) mis-detect the dark ironwork as shadows. The last three rows show shadow detection results on some challenging cases, where true shadows are visually similar to their context (potential false negatives). We can see that our method can still successfully disambiguate these cases, while other methods miss some true shadow regions. For example, in the fourth row, all existing methods, except BDRAR, detect some part(s) of the white strip under the shadow regions as non-shadow (false negative), while BDRAR fails to detect the non-shadow region between the two legs. In contrast, our method can predict them correctly.

Finally, we would like to study the FP and FN predictions produced by our DS module, as shown in Figure 5. These results can shed light on how explicitly integrating the distraction semantics can help shadow detection, particularly for some challenging cases. For example, in the top row, our FP predictor estimates the black parts of the pillar on the road as false positives, which helps our model discriminate that region from the shadow correctly. In the second row, our FN predictor activates on the shadow region with high visual similarity to its surrounding buildings. This can help our model resolve possible ambiguities within the shadow region.

4.4. Comparison with Salient Object Detection Methods

Salient object detection methods take an image as input and output a dense label prediction map. Such models can be retrained on the shadow detection datasets and used for shadow detection. For comparison, we re-train and test two recent salient object detection models [1, 35] on the

Figure 4. Qualitative results of our method, compared with other shadow detection methods.

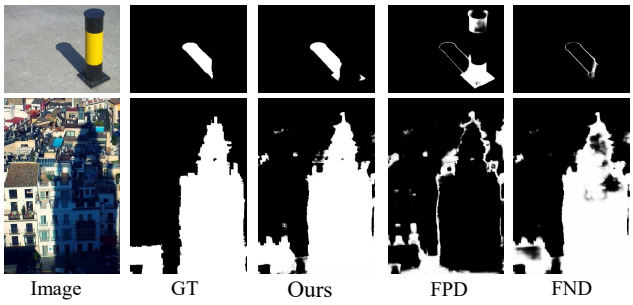

Figure 5. Visual results of our shadow and distraction detection.

same datasets as ours. We re-train these models using the codes from the authors and adjust the parameters for best performances. The last two rows of Table 1 report the results. We see that salient object detection models can give promising results on shadow detection, and in some situations, they perform even better than some of the shadow detection methods [34, 32, 24]. However, our method still outperforms them.

4.5. Effects of Distraction Semantics

To explore the effectiveness of considering distraction semantics explicitly in our model, we compare our full model with its several variants: (1) without the DS module or DS loss: we remove the DS module and the DS loss in Eq. 3. This can be considered as a plain baseline. (2) with only the DS module. (3) with only the DS loss. The results are reported in Table 2. Without the DS module or distraction-aware loss, our model has the worst performance. Adding either the DS module or DS loss will lead to a large performance improvement. When both DS module and DS loss are used, our model (full model) achieves the best performance. This suggests that explicitly considering distractions is crucial to our performance, and both DS module and DS loss are necessary in our model.

	SBU	UCF	ISTD
w/o DS module and DS loss	4.42	8.50	3.41
w/ only DS module	3.62	7.80	2.60
w/ only DS loss	3.89	8.08	2.50
w/o FN Sub-module	3.71	7.63	2.57
w/o FP Sub-module	3.68	8.43	2.41
Simple fusion	3.79	7.68	2.58
Both attentions	3.66	8.41	2.38
Both convs	3.71	7.79	2.49
Full model (Ours)	**3.45**	**7.59**	**2.17**

Table 2. Ablation study results. BER scores are reported on three datasets. Best results are highlighted in bold.

4.6. Ablation Study

To evaluate the design choices of the proposed distraction-aware shadow module, we compare our DS

	SBU	UCF	ISTD
No supervision	3.84	7.60	2.23
Single model	3.75	8.73	2.74
Our model	4.04	8.37	2.90
Multiple models	**3.45**	**7.59**	**2.17**

Table 3. Results of different strategies for generating distraction supervision. BER scores are reported on three datasets. Best results are highlighted in bold.

module with its ablated versions:

- Simple fusion: We replace the DS modules with a simple fusion method, where we add the FN features to the image features and subtract the FP features from the resulting features to get the DS features.
- Both attentions: We set the architecture of the FP sub-module the same as that of the FN sub-module.
- Both convs: We set the architecture of the FN sub-module the same as that of the FP sub-module.
- w/o FN Sub-module: We remove the FN sub-module and only keep the FP Sub-module.
- w/o FP Sub-module: We remove the FP sub-module and only keep the FN Sub-module.

In Table 2 (4th-8th rows), we can see that the simple fusion method has the worst performance, suggesting that direct fusion with simple addition is not effective in incorporating distraction semantics. When only one type of distraction is considered (4th-5th rows), the performance is worse than our full model, showing the necessity of considering both types of distractions simultaneously. When the same architecture is used for FN and FP sub-modules (7th-8th rows), the performance is also worse than our full model. This implies that it is important to develop different fusion strategies to accommodate the distinctive characteristics of the two types of distractions.

4.7. Distraction Supervision Generation Strategies

We further explore the effect of different strategies for generating distraction supervision, as discussed in Section 3.2:

- No supervision: We remove the distraction supervision and only use the shadow maps as supervision (Eq. 2 and 3) to learn our model.
- Single model: We only choose a baseline model [10] to compute the distraction supervision.
- Our model: After our model is trained, we use the distractions generated by our trained model as supervision to re-train our model.
- Multiple models: We combine the outputs of multiple existing models, as used in our full model.

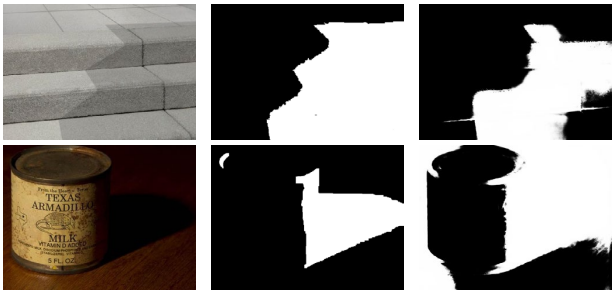

Figure 6. Failure cases. Our method may fail on some weak shadow images (top) or some images with very dark background (bottom).

Table 3 shows the results on three datasets, SBU, UCF and ISTD. Using multiple models has better performance than without using any distraction supervision or using only a single model. Without using any distraction supervision signals, it will be difficult for the model to learn useful distraction-aware features. Our strategy of combining multiple models can alleviate the bias of a single model, allowing us to generate more reliable distraction supervision for our model. Note that generating distraction supervision using our trained model gives the worst performance. It may be that since the distractions generated by our trained model become very rare cases, due to its superior performance in dealing with FPs and FNs, forcing our model to focus on the few rare cases would compromise its performance on other common cases, resulting in the decrease in the overall performance.

5. Conclusion

In this paper, we have proposed a Distraction-aware Shadow Detection Network (DSDNet) for shadow detection. Our network explicitly considers distraction semantics via a proposed Distraction-aware Shadow (DS) module. The DS module augments input image features with explicitly learned distraction features via a carefully designed fusion strategy to produce distraction-aware features for robust shadow detection. Experimental results show that our model can resolve challenging and ambiguous cases in shadow detection favorably, yielding new state-of-the-art performances on the SBU, UCF and ISTD datasets.

Although our method can handle most challenging cases, it may fail on some weak shadow images (where the shadows have very similar brightness to the background) or some images with very dark background (where the shadows are almost blended into the background), as shown in Figure 6. A possible solution to this problem is to augment the dataset with more weak shadow samples or dark background samples. As a future work, we plan to look into various types of extreme shadow problems.

References

[1] S. Chen, X. Tan, B. Wang, and X. Hu. Reverse attention for salient object detection. In *ECCV*, 2018. 3, 6

[2] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *ITSC*, 2001. 1

[3] G. D. Finlayson, M. S. Drew, and C. Lu. Entropy minimization for shadow removal. *IJCV*, 2009. 1, 2

[4] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew. On the removal of shadows from images. *TPAMI*, 2006. 1, 2

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[6] Y.-P. Guan. Wavelet multi-scale transform based foreground segmentation and shadow elimination. *The Open Signal Processing Journal*, 2008. 1

[7] R. Guo, Q. Dai, and D. Hoiem. Paired regions for shadow detection and removal. *TPAMI*, 2013. 1, 2

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[9] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 1998. 2

[10] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. In *CVPR*, 2017. 8

[11] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. *TPAMI*, 2018. 3

[12] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P.-A. Heng. Direction-aware spatial context features for shadow detection. In *CVPR*, 2018. 1, 3, 4, 5, 6

[13] Q. Huang, C. Xia, C. Wu, S. Li, Y. Wang, Y. Song, and C.-C. J. Kuo. Semantic segmentation with reverse attention. In *BMVC*, 2017. 3

[14] X. Huang, G. Hua, J. Tumblin, and L. Williams. What characterizes a shadow boundary under the sun and sky? In *ICCV*, 2011. 1, 2

[15] I. N. Junejo and H. Foroosh. Estimating geo-temporal location of stationary cameras using shadow trajectories. In *ECCV*, 2008. 1

[16] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. *TOG*, 2011. 1

[17] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri. Automatic shadow detection and removal from a single image. *TPAMI*, 2016. 1

[18] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 5

[19] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Estimating natural illumination from a single outdoor image. In *ECCV*, 2009. 1

[20] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *ECCV*, 2010. 2

[21] H. Le, T. F. Y. Vicente, V. Nguyen, M. Hoai, and D. Samaras. A+ D net: Training a shadow detector with adversarial shadow attenuation. In *ECCV*, 2018. 1, 2, 4, 5, 6

[22] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[23] I. Mikic, P. C. Cosman, G. T. Kogut, and M. M. Trivedi. Moving shadow and object detection in traffic scenes. In *ICPR*, 2000. 1

[24] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, and D. Samaras. Shadow detection with conditional generative adversarial networks. In *ICCV*, 2017. 1, 2, 6, 7

[25] T. Okabe, I. Sato, and Y. Sato. Attached shadow coding: Estimating surface normals from shadows under unknown reflectance and lighting conditions. In *ICCV*, 2009. 1

[26] A. Panagopoulos, D. Samaras, and N. Paragios. Robust shadow and illumination estimation using a mixture model. In *CVPR*, 2009. 1, 2

[27] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios. Illumination estimation and cast shadow detection through a higher-order graphical model. In *CVPR*, 2011. 1

[28] L. Qu, J. Tian, S. He, Y. Tang, and R. W. Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *CVPR*. 1

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[30] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *CVIU*, 2004. 1

[31] L. Shen, T. Wee Chua, and K. Leman. Shadow optimization from structured deep edge detection. In *CVPR*, 2015. 2

[32] T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, and D. Samaras. Large-scale training of shadow detectors with noisily-annotated shadow examples. In *ECCV*, 2016. 1, 2, 5, 6, 7

[33] Y. Vicente, F. Tomas, M. Hoai, and D. Samaras. Leave-one-out kernel optimization for shadow detection. In *ICCV*, 2015. 2, 5

[34] J. Wang, X. Li, and J. Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *CVPR*, 2018. 1, 2, 6, 7

[35] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, 2017. 6

[36] Y. Wang, X. Zhao, Y. Li, X. Hu, K. Huang, and N. CRIPAC. Densely cascaded shadow detection network via deeply supervised parallel fusion. In *IJCAI*, pages 1007–1013, 2018. 3

[37] L. Wu, X. Cao, and H. Foroosh. Camera calibration and geo-location estimation from two shadow trajectories. *CVIU*, 2010. 1

[38] H. Xiao, J. Feng, Y. Wei, M. Zhang, and S. Yan. Deep salient object detection with dense connections and distraction diagnosis. *TMM*, 2018. 3

[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 3

[40] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 3

[41] J. Zhu, K. G. Samuel, S. Z. Masood, and M. F. Tappen. Learning to recognize shadows in monochromatic natural images. In *CVPR*, 2010. 1, 2, 5

[42] L. Zhu, Z. Deng, X. Hu, C.-W. Fu, X. Xu, J. Qin, and P.-A. Heng. Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection. In *ECCV*, 2018. 1, 3, 4, 5, 6

[43] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*. 3