Instance Shadow Detection with A Single-Stage Detector

Tianyu Wang, Xiaowei Hu*, Pheng-Ann Heng, and Chi-Wing Fu

Abstract—This paper formulates a new problem, instance shadow detection, which aims to detect shadow instance and the associated object instance that cast each shadow in the input image. To approach this task, we first compile a new dataset with the masks for shadow instances, object instances, and shadow-object associations. We then design an evaluation metric for quantitative evaluation of the performance of instance shadow detection. Further, we design a single-stage detector to perform instance shadow detection in an end-to-end manner, where the bidirectional relation learning module and the deformable maskIoU head are proposed in the detector to directly learn the relation between shadow instances and object instances and to improve the accuracy of the predicted masks. Finally, we quantitatively and qualitatively evaluate our method on the benchmark dataset of instance shadow detection and show the applicability of our method on light direction estimation and photo editing.

Index Terms—Instance shadow detection, instance segmentation, shadow detection, deep neural network.

INTRODUCTION 1

"When you light a candle, you also cast a shadow,"-Ursula K. Le Guin written in A Wizard of Earthsea.

Shadows are formed when the light is blocked by the objects. When we see a shadow, we also know that there must be some objects that create or cast the shadow. However, recent shadow detection methods [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] simply generate a binary mask to indicate the shadow regions and fail to find the associated object that casts each individual shadow. To find shadows together with their associated objects, we propose a new task, named instance shadow detection, in which we detect not only individual shadow instances in the input image but also the associated object that casts each shadow.

Instance shadow detection has the potential to benefit various applications. For privacy protection, for example, when we remove vehicles and persons from photos, we can remove the associated shadows with the objects together. Also for photo editing, when we translate or scale objects in photos, we can naturally manipulate the objects with their associated shadows simultaneously. Further, shadow-object associations give hints to estimate the light direction, facilitating the development of applications such as shadow generation for virtual objects in AR environments and scene relighting. Last, the detected shadow and object instances help to estimate building heights from satellite metadata [14].

To approach the new task of instance shadow detection, we first prepared the SOBA (Shadow OBject Association) dataset. The dataset has three parts: SOBA training, SOBA testing, and SOBA challenge. Both the SOBA-testing and SOBA-challenge sets are for testing but the SOBA-challenge set contains complex

- T. Wang and C.-W. Fu are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong and the Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong.
- X. Hu is with the Shanghai AI Laboratory.
- P.-A. Heng is with the Department of Computer Science and Engineering. The Chinese University of Hong Kong.
- The preliminary versions of this work were accepted for presentation in CVPR 2020 [1] and CVPR 2021 [2].
- Corresponding author: Xiaowei Hu (huxiaowei@pjlab.org.cn).



(c) Shadow instances

(e) Shadow-obj. associations

Fig. 1: Given (a) an input photo, the goal of instance shadow detection is to detect (c) individual shadow instances, (d) individual object instances, and (e) shadow-object associations. (b) shows the overall result produced by our method from (a).

scenarios for evaluating the capability of methods in handling challenging cases. The whole dataset contains 4,293 pairs of annotated shadow-object associations over 1,100 images. Each image has (i) a shadow instance mask, which labels each shadow instance with a unique color; (ii) a shadow-object association mask, which labels each shadow-object pair with a corresponding unique color; and (iii) an object instance mask, which is (ii) minus (i); see Figure 1 for an example. Also, we formulate a new evaluation metric for the task to quantitatively evaluate the performance of the instance shadow detection results.

We approach the instance shadow detection task by exploiting the remarkable computational capability of deep neural networks. Our earlier work LISA [1] first generates region proposals that likely contain shadow/object instances and shadow-object associations. For each proposal, we then crop regions-of-interest (RoIs) from the feature maps and predict masks and boxes of the shadow instances, object instances, and shadow-object associations from each RoI. Lastly, we pair the shadow and object instances with the shadow-object associations. However, this two-stage framework and post-processing strategy have several limitations. First, this approach considers shadow-object association as a single category. Yet, the appearance of shadow and object instances have large variations, so shadow-object associations could easily be missed. Second, it generates region proposals for shadow/object instances and shadow-object associations using two separate branches and leverages post-processing to produce the final shadow-object associations. Errors could accumulate over the network and postprocessing. Third, the employed RoIs represent feature regions using rectangular shapes. However, the shapes of the shadow instances and shadow-object associations are usually irregular and the cropped RoIs of rectangular shapes could include many irrelevant image contents such as other object and shadow instances.

To address the above issues, we design a single-stage deep framework [2] to directly learn to find the relation between shadow and object instances in an end-to-end manner. This framework includes only fully convolutional operations to generate the masks for the shadow instances, object instances, and shadow-object associations, thus enabling us to handle shadow/object instances and shadow-object associations of any shape. Importantly, we design the bidirectional relation learning module to find the shadow-object association pairs to learn an offset vector from the center of each shadow instance to the center of its associated object instance, and the other way around, aiming to explore the inter-relationship between shadows and objects effectively. We construct a class vector to represent the learning directions during this process: shadow to object or object to shadow.

Further, we design a deformable MaskIoU head in the network to improve the mask accuracy. This module takes the output of the mask head and mask feature as inputs and predicts the IoU scores of the predicted masks. Unlike the Mask Scoring R-CNN [15], which feeds feature cropped from the RoI as the input of the MaskIOU head, we take the raw feature as input to our method. Hence, we further introduce the deformable convolution [16] to process the whole feature and focus on discriminate regions in the shadow/object instance masks. Also, we formulate a segmentation loss, an offset loss, and a boundary loss to jointly optimize the entire network. Lastly, we design also a shadow-aware copy-andpaste strategy to augment input images during the training. These new techniques help the network learn to better pair the shadow and object instances for improving the overall performance.

Below, we summarize the major contributions of this work.

- First, we formulate a new task, instance shadow detection, which aims to find individual shadow instances, individual object instances, and the shadow-object associations.
- Second, we prepare a new dataset and evaluation metric to support instance shadow detection. The dataset contains 1,100 images and 4,293 pairs of shadow-object associations, and provides three instance masks for each image.
- Third, we design a single-stage instance shadow detection network with two novel techniques, the bidirectional relation learning module, the deformable maskIoU head, and some training strategies to directly learn the relation between shadow and object instances.
- Fourth, we perform various experiments to quantitatively and visually demonstrate the effectiveness of our method. Results show that our method outperforms our previous two-stage detector [1] by over 50.2% and 70.2% on the SOBA-testing set and the SOBA-challenge set, respectively.
- Last, we demonstrate the applicability of the instance shadow detection on various tasks, including light direction estimation and photo editing.

2 RELATED WORK

Shadow detection. Generic shadow detection aims to generate a binary mask to mark shadow regions in the input image. Early methods build physical models to leverage color and illumination to detect shadows. Among them, Salvador *et al.* [17] explore shadows' spectral and geometrical properties to segment the cast shadows. Panagopoulos *et al.* [18] build a higher-order Markov Random Field illumination model with coarse 3D geometry information. Tian *et al.* [19] adopt the difference of spectral power distributions in daylight and skylight for shadow detection.

Later, machine-learning approaches are developed to recognize shadows by first describing image regions using handcrafted features and then classifying the regions into shadows and non-shadows. Features like texture [20], [21], [22], [23], Tjunction [24], color [21], [22], [23], [24], and edge [20], [24], [25] are commonly used to describe shadows followed by classifiers like SVM [21], [22], [23], [25] and decision tree [20], [24]. These designed physical models and hand-crafted features have limited ability to describe shadows, so approaches based on these models and features may mis-detect shadows in general cases.

Deep neural networks automatically learn features from shadow images and show remarkable performance on shadow detection, especially with extensive training data. Khan *et al.* [7] present the first work that uses a convolutional neural network (CNN) to learn features for shadow detection. Shen *et al.* [26] and Hou & Vicente *et al.* [3], [10] devise a structured learning framework and a stacked-CNN, respectively, to detect shadows. Nguyen *et al.* [27] design an adjustable parameter in a conditional GAN to balance the weights of shadow and non-shadow regions.

Later, Hu et al. [5], [6] learn the direction-aware spatial context to detect shadows by designing an attention mechanism in a spatial recurrent network. Wang et al. [28] iteratively detect and remove shadows with two conditional generative adversarial networks. Le et al. [8] adopt adversarial training samples generated from a shadow attenuation network to train a shadow detection network. Zhu et al. [12] design a bidirectional feature pyramid network with recurrent attention residual modules to detect shadows. Zheng et al. [11] revisit false negatives and false positives from the predicted results and derive a distraction-aware shadow detection network. Ding et al. [29] detect and remove shadows in a recurrent manner via an attentive recurrent generative adversarial network. More recently, Chen et al. [13] present a semi-supervised shadow detection algorithm by exploring unlabeled data through a multitask mean teacher framework. Hu et al. [4] build a new dataset to support shadow detection in a complex world and designed a fast shadow detection network. Chen et al. [30] design a triplecooperative video shadow detection network to detect shadows in videos. Unlike general shadow detection, which adopts a single mask for all shadows in an image, this work detects not just individual shadows but also the associated objects altogether.

Apart from generic shadow detection, various works explored deep learning to remove shadows in natural images [9], [29], [31], [32], [33], [34], [35], [36], [37], [38], [39] and in documents [40], to generate shadows in augmented reality [41] and in real scenes [42], and to manipulate portrait shadows [43]. Our instance shadow detection task offers a new perspective to edit or remove individual shadows with the associated objects.

Instance segmentation. Besides, this work relates to instance segmentation, which aims to label pixels of individual foreground objects in the input image. One category of methods predicts



(a) Example images in the SOBA-training set



(b) Example images in the SOBA-testing set

(c) Example images in the SOBA-challenge set

Fig. 2: Example images with the mask labels in our SOBA data set. Please zoom in for a better visualization.

region proposals in the input image and then generates an instance mask for each proposal, *e.g.*, MNC [44], DeepMask [45], Instance-FCN [44], SharpMask [46], FCIS [47], BAIS [48], MaskLab [49], Mask R-CNN [50], PANet [51], MegDet [52], and HTC [53]. Among them, Mask R-CNN simultaneously predicts the category label, bounding box, and segmentation mask for each region proposal and achieves great success. The other category directly predicts the instance masks and associated categories in the whole image, *e.g.*, TensorMask [54], SSAP [55], SOLO [56], Embed-Mask [57], SOLOv2 [58], CenterMask [59], and CondInst [60]. Our method is based on the architecture of CondInst [60]; below, we further elaborate on how CondInst works.

Details on CondInst [60]. CondInst performs instance segmentation by generating the location and mask of each object. First, it adopts a fully convolutional network to predict the object centers based on the features extracted by the backbone network. Second, it takes the object locations and the extracted mask features as inputs, and leverages the dynamic convolution to generate filters for each object to predict its mask. By doing so, CondInst can eliminate the RoI operations and reduce the parameters and computational complexity when predicting the masks, leading to a more efficient and simple instance segmentation framework. Based on CondInst, we further formulate our bidirectional relation learning module to learn the relation between shadow and object instances and design a deformable maskIoU head to penalize the predicted instance masks with low quality.

Difference from the conference papers. This work extends our earlier works [1], [2] in three aspects. First, we enrich our dataset prepared for instance shadow detection by providing more challenging cases with labels, aiming to evaluate the detection performance in complex scenarios. Second, we improve the singlestage instance shadow detection (SSIS) method (in our conference version [2]) by designing new techniques: (i) a deformable MaskIoU head, (ii) a shadow-aware copy-and-paste data augmentation strategy, and (iii) a boundary loss, to better segment the shadow/object instances and shadow-object associations. Our SSISv2 outperforms the two-stage detector LISA [1] and the original SSIS [2] by 50.6% and 17.2%, respectively, in accuracy. Third, we perform more experiments to evaluate the design of our network and add more applications to show how our SSISv2 outperforms the existing methods on instance shadow detection.

3 DATASET AND EVALUATION METRIC

3.1 SOBA (Shadow OBject Association) Dataset

We prepare SOBA (Shadow OBject Association) dataset to support instance shadow detection, which contains three parts: SOBA training, SOBA testing, and SOBA challenge. We first build the SOBA-training and -testing sets from relatively simple cases by collecting images from the ADE20K [61], [62], SBU [3], [10], [63], ISTD [28], and Microsoft COCO [64] datasets, and also



(b) Statistical properties of the SOBA-challenge set.

Fig. 3: Statistical properties of the SOBA dataset.

from the Internet using keyword search with shadow plus animal, people, car, athletic meeting, zoo, street, etc. Then, we coarsely label the images to produce shadow instance masks and shadow-object association masks, and refine them using the Apple Pencil software; see Figures 1 (b) & (d). Next, we obtain object instance masks (see Figure 1 (c)) by subtracting each shadow instance mask from the associated shadow-object association mask. Overall, there are 1,000 images with 3,623 pairs of shadow-object instances, and we randomly split the images into a training set (840 images, 2,999 pairs) and a testing set (160 images, 624 pairs); see Figure 2 (a) & (b) for some examples.

We show some statistical properties of the SOBA-training and -testing sets in Figure 3 (a). From the left histogram, we can see that it has a diverse number of shadow-object pairs per image, around 3.62 pairs per image on average. On the other hand, the right histogram reveals the proportion of image space (horizontal axis) occupied, respectively, by shadow and object instances in the dataset images. From the plot, we can see that most shadows and objects occupy relatively small areas in the whole images.

To evaluate the detection performance in complex scenarios, we further collected 100 images with challenging shadow-object pairs from the Internet using keyword search with *crowd* plus people, animals, cars, street, pasture, grassland, and beach. Then, we picked images of scenes with multiple various-shape shadow-object associations, large occlusion between the objects, between the shadows, or between both the objects and shadows, and long shadows that usually appear at sunset. Figure 2 (c) shows some of these images. Also, we annotated the images using similar steps as mentioned earlier. We name this dataset SOBA challenge, which includes 670 pairs of shadow-object instances, and the whole SOBA-challenge dataset is used only for testing.

Figure 3 (b) shows SOBA challenge's statistical properties. From the left histogram, we can see that this dataset has ~ 6.70 pairs per image on average (vs. 3.62 for SOBA training & testing) and more than 20% of the images have nine or more shadow-object pairs per image. The right histogram also shows that this dataset contains more objects that occupy large areas in the images.

Overall, the whole SOBA dataset has 1,100 images with 4,293 pairs of annotated shadow-object associations.

3.2 SOAP (Shadow-Object Average Precision) Metric

Existing metrics evaluate instance segmentation results by looking at object instances individually. Our problem involves multiple types of instances: shadows, objects, and their associations. Hence, we formulate a new metric called the *Shadow-Object Average Precision* (SOAP) by adopting the same formulation as the traditional average precision (AP) with the intersection over union (IoU) but further considering a sample as true positive (an output shadowobject association), if it satisfies the following three conditions:

- (i) the IoU between the predicted shadow instance and groundtruth shadow instance is no less than τ ;
- (ii) the IoU between the predicted object instance and groundtruth object instance is no less than τ ; and
- (iii) the IoU between the predicted and ground-truth shadowobject associations is no less than τ .

We follow [64] to report the evaluation results by setting τ as 0.5 (SOAP₅₀) or 0.75 (SOAP₇₅), and report also the average over multiple τ [0.5:0.05:0.95] (SOAP). Also, since we obtain the bounding boxes and masks of the shadow instances, object instances, and shadow-object associations, we report SOAP₅₀, SOAP₇₅, and SOAP for both bounding boxes and masks. *The dataset and evaluation metric are available for download at https://github.com/stevewongv/InstanceShadowDetection.*

4 METHODOLOGY

4.1 Overall Network Architecture

Figure 4 overviews our network architecture. Given the input image, we leverage a convolutional neural network to extract feature maps in varying solutions and employ a feature pyramid network [65] with multiple feature levels (P3 to P7). Then, we adopt multiple heads at different levels: a class tower with four convolutional layers to predict the classification scores and a box tower with another four convolutional layers for other predictions. In summary, we obtain the following predictions for each head:

- (i) classification scores, which indicate the categories of shadow, object, and background;
- (ii) offset vector, which are image-space vectors from the centers of shadow instances to the centers of the corresponding object instance, and vice versa;
- (iii) controller and paired controller, each learning a set of filter parameters in the mask head to predict the masks for shadow instance and object instance, respectively. Note that each instance has its individual filter parameters to predict a mask; see [60] for details. In our framework, if the controller generates filter parameters for a shadow instance, the paired controller will generate filter parameters for the corresponding object instance, and vice versa; and
- (iv) regression and centerness: regression predicts the bounding box of each shadow and object instance, whereas centerness regularizes the prediction by reducing the number of lowquality predicted bounding boxes far from the center of a target shadow/object; see [66] for details.

Next, we formulate a mask branch, which takes the feature map at P3 as input and generates the mask feature. For each predicted shadow/object instance, we duplicate and concatenate the mask feature with two relative coordinate (Rel. Coord.) maps:



Fig. 4: The schematic illustration of our single-stage instance shadow detection network (SSISv2). The mask feature and outputs of the *box tower* and *class tower* are used to formulate the bidirectional relation learning module; see Figure 5. The mask feature and output instance masks are sent to the *deformable MaskIoU head* for mask refinement; see Section 4.3. Note that each head has its own box head and class head, and the filter parameters among these heads are shared.

one indicates the center of the object/shadow instance, whereas the other is obtained by first multiplying the offset vector with a class vector then adding the results with the coordinates to represent the center of the corresponding shadow/object instance. Note that the class vector is generated from the classification score, where -1 (+1) indicates the direction from object to shadow (from shadow to object) and the relative coordinate map is computed from the predicted locations of shadow/object instances. Further, we use the learned filter parameters from the controller and paired controller to perform convolutional operations on the concatenated feature mask and relative coordinate maps and predict the masks for the shadow/object instances. Finally, we concatenate the predicted masks for the instances and the mask feature and design a deformable MaskIoU head to refine the predicted masks by adopting a MaskIoU loss function.

In the following, we will elaborate on how to learn the relation between shadow and object instances (Section 4.2) and formulate the deformable MaskIoU head (Section 4.3), and then present the training and testing strategies, including the shadow-aware copyand-paste augmentation and loss functions (Section 4.4).

4.2 Bidirectional Relation Learning

Figure 5 shows the detailed structure of our proposed bidirectional relation learning module. Figure 5 (a) illustrates how to learn the paired shadow instance from the object instance, whereas Figure 5 (b) illustrates this strategy in the opposite direction. As shown in the top left corner, after obtaining the original location L^m of the *m*-th object instance, we append the location with the mask

feature and adopt the m-th mask head to predict the segmentation mask of this instance. Note that the filter parameters in the mask head are produced from the controller and the filter parameters vary in different mask heads; see "Controller" in Figure 4.

Then, we compute the associated location A^m to mark the center of the paired shadow instance by using the learned offset vector O^m and class vector -1:

$$A^m = L^m + O^m \times -1, \qquad (1)$$

where the offset vector is learned from the box tower and it represents the distance between the center of the object instance and the center of the paired shadow instance; the class vector is generated from the classification score and we adopt -1 to represent the direction from object to shadow and 1 to represent the direction from shadow to object. Next, we concatenate the associated location A^m and mask feature, and use the *m*-th associated mask head to generate the mask for the shadow instance, and the filter parameters of the associated mask head are learned from the paired controller automatically, as shown in Figure 4.

Similarly, taking the original location L^n of the *n*-th shadow instance as the input, we compute the associated location A^n of the paired object instance by

$$A^n = L^n + O^n \times 1, \qquad (2)$$

where O^n denotes the *n*-th offset vector and 1 denotes the learning direction from shadow to object. Also, we adopt the mask head and the associated mask head to generate the segmentation masks for the paired shadow and object instances; see Figure 5 (right).

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE



Fig. 5: The schematic illustration of the bidirectional relation learning module in our network. The left part (Object \rightarrow Shadow) shows how to find the associated shadow instance from the location of the paired object instance, whereas the right part (Shadow \rightarrow Object) shows how to find the associated object instance from the location of the paired shadow instance.



Fig. 6: Confidence scores vs. mask IoUs before and after applying the MaskIoU head or Deformable MaskIoU Head. Each point in the figure denotes a predicted instance mask. The thick lines in the plots indicate Confidence score equals mask IoU. As shown in (c), with the Deformable MaskIoU head, we can dramatically avoid more masks with high confidence scores but low IoUs.

Note that the location maps (L^m, A^m, L^n, A^n) shown in Figure 5 are the visualization results of the learned locations, demonstrating that our network can successfully learn the locations for the shadow and object pairs.

4.3 Deformable MaskloU Head

As shown in Figure 6 (a), the original model tends to predict masks with high confidence scores but low IoUs. These low-quality masks degrade the detection performance, since the confidence score predicted from the classification task ("classification" in Figure 4) fails to consider the mask information. To further refine the predicted masks of shadow/object instances, we formulate the deformable MaskIoU head to regress the intersection over union (IoU) between the predicted masks and the associated ground-truth masks. As shown in Figure 4 (bottom), given the mask feature and each predicted mask as input, we perform a 1×1 convolution to reduce the feature channel and a deformable convolution layer [16] to focus the learning on the instance's specific region, followed by a convolution layer and an adaptive max-pooling layer to reshape the feature map to 64×64 . Lastly, we leverage three fully connected layers to predict a single mask IoU per instance.

Unlike the MaskIoU head in [15], which is designed only for RoI-based methods and takes the RoI feature of size 14×14 as input, we design a deformable MaskIoU head that takes the whole mask feature with instance mask as input and automatically learns the discriminative feature for each instance mask, since our based method [60] employs the whole mask feature with conditional convolution to eliminate the RoI operations. Figure 6 shows the predicted instance masks with confidence scores and associated mask IoUs before and after using the MaskIoU head. As shown on the bottom right of the figure, after using the deformable MaskIoU head, we can dramatically avoid more masks with high confidences but low IoUs, showing that our deformable MaskIoU head can successfully filter out instances of low quality. Please see Section 5.2 for related quantitative comparison results.

4.4 Training and Testing Strategies

4.4.1 Shadow-aware Copy-and-Paste Augmentation

To enhance the network's robustness, especially for handling challenge cases, *e.g.*, occlusion between object and shadow instances, we design a shadow-aware copy-and-paste augmentation strategy to enrich the training data. As shown in Figure 7, we IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE



Fig. 7: Shadow-aware copy-and-paste augmentation. (b) & (c) show example copying-and-pasting results on different objects.

randomly select a shadow-object association in the input image, copy the object instance with its associated shadow instance, then paste them to the surroundings. Specifically, we shift the mask by a random value in range $[-\frac{2}{3}W, \frac{2}{3}W]$ on the X axis and a random value in range $(0, \frac{2}{3}H]$ on the Y axis, where W and H are the width and height of the shifted object. Importantly, the augmentation should consider *object layering*. That is, we should put the pasted shadow-object association behind existing object instances but above their original shadow instances and scene background for plausible occlusions among the objects. Further, we propose to relight the scene background in the shadow region cast by the copied object. The relighted shadow region R is computed by

$$R = \frac{mean(S)}{mean(T)} \cdot T , \qquad (3)$$

where T is the original color of the relighted shadow region and S is the color of the shadow region where is copied.

4.4.2 Loss Function

We define the overall loss \mathcal{L}_{all} for training our SISSv2 network as a sum of detection loss \mathcal{L}_D , mask loss \mathcal{L}_M , and boundary loss \mathcal{L}_B :

$$\mathcal{L}_{all} = \mathcal{L}_{D} + \mathcal{L}_{M} + \mathcal{L}_{B} . \tag{4}$$

Detection loss:

$$\mathcal{L}_{\rm D} = \mathcal{L}_{\rm cls} + \mathcal{L}_{\rm center} + \mathcal{L}_{\rm box} + \mathcal{L}_{\rm offset}, \tag{5}$$

where \mathcal{L}_{cls} is the classification loss, \mathcal{L}_{center} is the centerness loss, and \mathcal{L}_{box} is the box regression loss, which follows the losses in [66]. The offset loss \mathcal{L}_{offset} takes the form of the smooth \mathcal{L}_1 loss [67] for optimizing the offset vectors:

$$\mathcal{L}_{\text{offset}}(u, v) = \sum_{I \in \{x, y\}} \left\{ \begin{array}{ll} 0.5 \left(u_i - v_i\right)^2, & \text{if } |u_i - v_i| < 1; \\ |u_i - v_i| - 0.5, & \text{otherwise}, \end{array} \right.$$
(6)

where u_i is resulted from the element-wise multiplication of the predicted offset vector and class vector:

$$u_i = O_i \times C_i , \qquad (7)$$

and v_i denotes the ground-truth offset vector:

$$v_i = \hat{L}_i - L_i , \qquad (8)$$



Fig. 8: The mask head (top right) simultaneously predicts *a thick boundary map* and an instance mask. We then pass the instance mask to the Laplacian filter to produce the *thin boundary map*.

where \tilde{L}_i and L_i are the ground-truth and predicted location of the paired object/shadow instance, respectively.

Mask loss:

$$\mathcal{L}_{\rm M} = \mathcal{L}_{\rm mask} + \mathcal{L}_{\rm mask}^{\rm associated} + \mathcal{L}_{\rm maskiou} , \qquad (9)$$

where we adopt dice loss [68] to compute the losses of the output instance masks $\mathcal{L}_{mask}^{associated}$; see Figure 4 for the predictions. The MaskIoU loss $\mathcal{L}_{maskiou}^{associated}$; see Figure 4 for the predictions.

$$\mathcal{L}_{\text{maskiou}} = \frac{1}{N} \sum_{i=1}^{N} (I_i - \tilde{I}_i)^2 , \qquad (10)$$

where I_i and \tilde{I}_i are the predicted and ground-truth mask IoU, respectively; and N is the number of the predicted instances.

Boundary loss. Different from existing boundary losses, we formulate two types boundary maps with different thicknesses to improve the boundary accuracy of the instance masks. One is a thick boundary map for focusing on the boundary structures, and the other is a thin boundary map for focusing on the boundary details. As shown in Figure 8, we predict the thick boundary map from the mask head directly and generate the thin boundary map by applying a Laplacian filter on the predicted instance mask. On the other hand, we extract the boundary map from the groundtruth image and apply the Laplacian filter to the boundary map to formulate a supervision on the predicted thin boundary map; then, we apply the Euclidean distance transform [69] to the boundary map from the ground truth to formulate a supervision on the predicted thick boundary map. The overall boundary loss is a summation of the output instance masks $\mathcal{L}_{boundary}$ and the output associated instance masks $\mathcal{L}_{\text{boundary}}^{\text{associated}}$:

$$\mathcal{L}_{\rm B} = \mathcal{L}_{\rm boundary} + \mathcal{L}_{\rm boundary}^{\rm associated} , \qquad (11)$$

$$\mathcal{L}_{\text{boundary}} = \beta \frac{||l(\tilde{m})| - |l(m)||}{|l(\tilde{m})|} + dice(\frac{d(\tilde{m})}{max(d(\tilde{m}))} < 0.5, b) ,$$
(12)

where \tilde{m} is ground-truth instance mask; m is predicted instance mask; l(x) is Laplacian filter, whose kernel size is five; weight β is set as five to balance the loss values; d computes a distance field, in which each pixel stores the distance to the nearest boundary pixel; $max(d(\tilde{m}))$ is the maximum distance for normalization; dice is dice loss; and b is the predicted thick boundary map.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

TABLE 1: Comparison with the previous state-of-the-art methods for instance shadow detection on the SOBA-testing set. Note that the results of LISA and SSIS are slightly different from the results reported in the conference versions [1], [2], because this work replaces a simple polygon representation (employed in the previous works) with a more precise representation, *i.e.*, RLE, for the labeled masks.

Network	$SOAP_{segm}$	$SOAP_{bbox}$	Association AP_{segm}	Association AP_{bbox}	Instance AP_{segm}	Instance AP_{bbox}
LISA [1] SSIS [2]	23.5 30.2	21.9 27.1	40.9 52.2	48.4 59.6	39.2 43.4	37.6 41.3
SSISv2	35.3	29.0	59.2	63.1	50.2	44.4

TABLE 2: Comparison with the previous state-of-the-art methods for instance shadow detection on the SOBA-challenge set.

Network SOA	P_{segm} $SOAP_{bbox}$	Association AP_{segm}	Association AP_{bbox}	Instance AP _{segm}	Instance AP_{bbox}
LISA [1] 19	0.4 10.1	20.7	25.8	23.8	24.3
SSIS [2] 19	2.7 12.8	28.4	32.6	25.6	26.2
SSISv2 1	7.7 15.1	34.6	37.3	31.0	28.4

Figure 12 shows example results produced by our method with and without the boundary loss; by adopting the boundary loss in training, we can improve the accuracy of the predicted instance masks; please see Section 5.2 for the quantitative comparison. Both offset loss \mathcal{L}_{offset} and mask loss of the associated instance mask $\mathcal{L}_{mask}^{associated}$ propagate the gradient to offset vectors, helping to optimize the network during the training. Also, we do not use $\mathcal{L}_{maskiou}$ in the first 5,000 training iterations and thin boundary loss in the first 10,000 training iterations, as the predicted instance masks have low quality at the beginning of the training process.

4.4.3 Training Parameters

We train our network by adopting the strategies of CondInst [60] and AdelaiDet [70]. First, we adopt the weights of ResNeXt-101-BiFPN [71], [72] trained on ImageNet [73] to initialize the backbone network parameters, set the mini-batch size as two, and optimize our network on one NVidia RTX 3090 GPU. Second, we set the base learning rate as 0.001, adopt a warm-up [74] strategy to linearly increase the learning rate from 0.0001 to 0.001 in the first 1,00 iterations, reduce the learning rate to 0.0001 after 40,000 iterations, and stop the learning after 45,000 iterations. Third, we re-scale the input images, such that the longer side is smaller than 1,333 and the shorter side was smaller than 640, without changing the image aspect ratio. Lastly, we apply random horizontal flip to the input images as data augmentation.

4.4.4 Inference

In testing, the mask heads in our network produce the masks for the shadow and object instances, while the associated mask heads generate the masks for the paired object and shadow instances based on the learned offset vectors; see Figure 5. With bidirectional relation learning, we can obtain two sets of predicted masks, for each pair of shadow and object instances. If the main branch (left branch in Figure 5 (a)&(b)) produces the mask of its shadow instance, the associated branch (right branch in Figure 5 (a)&(b)) will generate the mask of its object instance, and vice versa. Yet, the accuracy of mask predictions in the main branch is usually better than that of the associated branch, since the associated branch needs to learn both tasks of mask prediction and shadow-object relation, making its training more difficult. Hence, we adopt the associated branch only to predict the paired relation of the shadow and object instances, and take the masks predicted from the main branch as the results. Finally, we adopt mask nonmaximum suppression (NMS) to refine the results.

5 EXPERIMENTAL RESULTS

5.1 Comparison with State-of-the-art Methods

We compare our SSISv2 with the instance shadow detection methods in our conference versions, LISA [1] and SSIS [2]. LISA is a two-stage detector that takes light direction as guidance and adopts a post-processing strategy to pair up the predicted shadow/object instances with the shadow-object associations, whereas SSIS is a single-stage fully convolutional detector that directly predicts shadow instances, object instances, and their associations. Our SSISv2 further formulates the deformable MaskIoU head, the shadow-aware copy-and-paste data augmentation strategy, and the boundary loss to improve the performance over SSIS [2].

Table 1 reports the comparison results on the SOBA-testing set. We can see that SSISv2 clearly outperforms previous stateof-the-art methods, LISA [1] and SSIS [2], for all the evaluation metrics, where the improvements on $SOAP_{segm}$ and $SOAP_{bbox}$ are 50.2% / 32.4% over LISA and 16.9% / 7.0% over SSIS, respectively, showing the superiority of SSISv2. Further, we compare the methods on the SOBA-challenge set and report the results in Table 2; SSISv2 also achieves the best results in terms of all the evaluation metrics in the challenge scenarios.

Next, we provide visual comparison results in Figure 9, where (a) shows the input images; (b), (c), and (d) show the results produced by LISA [1], SSIS [2], and SSISv2, respectively, and (e) shows the paired locations learned by SSISv2 to indicate the paired shadow and object instances. From the results, we can see that (i) SSISv2 can discover more shadow-object association pairs, as shown in the first two rows; (ii) SSISv2 can produce more accurate masks for shadow and object instances, as shown in the third row; (iii) SSISv2 can successfully pair up the object and shadow instances, but previous methods may fail; see the last two rows; and (iv) SSISv2 can learn the locations of shadow-object pairs through the directional relation learning module, as shown in (e). Figure 10 shows the visual comparison results on the SOBA-challenging set, where SSISv2 better pairs up the shadow and object instances and produces more accurate instance masks than the results produced by the previous methods. Please see Figure 11 for more instance shadow detection results produced by SSISv2 on various types of objects and shadows. Our code, trained models, and the results are released at https://github.com/stevewongv/SSIS.

5.2 Evaluation on the Network Design

Component analysis. We evaluate major components in SSISv2 on the SOBA-testing set. As shown in the first column in the



Fig. 9: Visual comparison between instance shadow detection results produced by various methods (b)-(d) on images (a) in the SOBAtesting set; (e) shows the learned locations for pairing shadow and object instances in our method.



Fig. 10: Visual comparison between instance shadow detection results produced by various methods (b)-(d) on the SOBA-challenge set.

TABLE 3: Component analysis on the SOBA-testing set; "data augm" denotes shadow-aware copy-and-paste (see Section 4.4.1).

	+ deformable maskIoU	+ boundary loss	+ data augm	$SOAP_{segm}$	$SOAP_{box}$	Association AP_{segm}	Association AP_{box}	Instance AP_{segm}	Instance AP_{box}
basic				28.1	26.8 25.4	49.8	56.6 57.6	41.5	40.8
+ class (SSIS [2])				30.2	27.1	53.6	59.6	43.4	41.3
	\checkmark			32.0	27.2	53.6	58.8	45.9	41.1
		\checkmark		31.0	26.3	54.6	59.4	45.0	40.8
			\checkmark	31.1	26.7	55.1	60.6	45.3	42.0
	\checkmark	\checkmark		33.3	27.5	56.0	60.7	46.4	42.0
	\checkmark		\checkmark	33.5	27.8	56.8	62.2	47.8	42.8
		\checkmark	\checkmark	32.1	27.7	56.4	61.7	46.3	42.8
SSISv2	\checkmark	\checkmark	\checkmark	35.3	29.0	59.2	63.1	50.2	44.4

TABLE 4: Evaluation on the bidirectional learning strategy.

Strategy	$SOAP_{segm}$	$SOAP_{bbox}$
$object \rightarrow shadow$	23.8	23.5
$shadow \rightarrow object$	25.6	23.1
main + associated	26.8	25.8
offset pairing	26.7	23.9
SSIS	30.2	27.2

TABLE 5: Evaluation on the MaskIoU head strategy.

Strategy	$SOAP_{segm}$	$SOAP_{bbox}$
w/o MaskIoU head w/o deformable conv	32.1 33.0	27.7 25.5
SSISv2	35.3	29.0

top half of Table 3, "basic" is a network built by removing the offset vectors, class vectors, deformable MaskIoU, shadow-aware copy-and-paste augmentation, and boundary loss from SSISv2 and adopting only the segmentation loss in training. "+ offset" learns the offset vectors based on the "basic" network. "+ class" further considers the class vectors, the same as the model in SSIS [2]. Table 3 (5-th to 11-th) rows show the new components in this work: "+ deformable MaskIoU" adopts the deformable MaskIoU head to refine the predicted masks; "+ data augm" adopts shadow-aware copy-and-paste augmentation; and "+ boundary loss" leverages the boundary loss to improve the boundary accuracy. Table 3 reports the analysis results, showing that all components consistently improve the performance for most metrics and best performance is attained when equipping all proposed components.

Bidirectional learning strategy analysis. Next, we evaluate the effectiveness of the bidirectional learning strategy. First, we learn the shadow-object pairs only in one direction. As shown in Table 4, for "object \rightarrow shadow," we used the architecture in Figure 5 (a) to predict the masks of the object instances from the mask heads in the main branch, and to predict the masks of the shadow instances from the associated heads. "shadow \rightarrow object" leverages the architecture in Figure 5 (b) for mask prediction. Then, we evaluate other strategies for finding the shadow-object associations. "main + associated" means we use the masks predicted from the main branch and the corresponding associated branch without using the strategy in Section 4.4.4-Inference. "offset pairing" means we replace the strategy in Section 4.4.4-Inference with the learned location offset between the shadow and object instances when pairing the association. Results show that learning the shadowobject relations from two directions with our inference strategy

TABLE 6: Evaluation on the boundary loss strategy.

Strategy	$SOAP_{segm}$	Association APsegm
w/o boundary loss	33.5	56.8
thick boundary loss	34.6	57.3
thin boundary loss	34.3	56.8
SSISv2	35.3	59.2

TABLE 7: Evaluation on shadow-aware copy-and-paste augm.

Strategy	$SOAP_{segm}$	$SOAP_{bbox}$
w/o data augm.	33.3	27.5
object-only	32.8	27.5
above layering	33.8	27.0
multiple associations	34.2	28.3
SSISv2	35.3	29.0

achieves the best performance.

MaskIoU head strategy analysis. To evaluate the effectiveness of our MaskIoU head design, we build two basic models: one by removing the MaskIoU head and the other by replacing the deformable convolution layers with naïve convolution layers. Results in Table 5 show that our MaskIoU head design with deformable convolution achieves the best performance.

Boundary loss strategy analysis. We quantitatively evaluate the effectiveness of the proposed boundary loss. Table 6 shows that both thin and thick boundary losses contribute to the performance and best performance is achieved by using both losses simultaneously; see also Figure 12 for visual comparison results.

Data augmentation strategy analysis. To evaluate the effectiveness of shadow-aware copy-and-paste data augmentation, we conduct experiments with different settings (see Table 7), where (i) "object-only" means we only copy and paste the object near its original position; (ii) "above layering" means we always put the pasted shadow-object association above the original object instance; (iii) "multiple associations" means we randomly select multiple objects from the image then copy and paste the corresponding shadow-object associations to their nearby positions. Table. 7 shows that (i) "object-only" decreases the performance compared with the baseline, since it lacks the information of the shadow instances and breaks the relations between shadows and objects; (ii) "above layering" hardly pastes the shadow naturally in front of the original object, thereby limiting the overall performance; and (iii) "multiple associations" introduces occlusions between associations, yet not as effective as SSISv2.

Discussion. SSISv2 has a strong ability of finding shadows and



Fig. 11: More instance shadow detection results produced by our SSISv2 over a wide variety of objects and shadows. The top two rows are from the SOBA-testing set while the others are from the SOBA-challenge set.



(a) w/o boundary loss

(b) w/ boundary loss

Fig. 12: Boundary loss analysis. (a) and (b) show the masks predicted from SSISv2 without and with boundary loss, respectively. We zoom into the regions in red boxes for better visualization.

objects. Yet, it is infeasible to handle some extreme scenarios, in which we cannot find another set of masks, e.g., very small shadows. In our implementation, we ignore instances that contain only one set of masks. In practice, this situation is very rare.

6 **APPLICATIONS**

Below, we present application scenarios to demonstrate the applicability of the results produced by our SSISv2.

Light direction estimation. Instance shadow detection promotes 2D light direction estimation in the image planes. For instance, we can connect the bounding box centers of the shadow and object instances in each shadow-object association pair as the estimated light direction. Figure 14 shows some example results, for which we adopt the estimated light directions to render virtual red posts with simulated shadows on the ground. From the results, we can see that the virtual shadows with the red posts [75] look consistent with the real shadows cast by the other objects, thus showing the applicability of our detection results of our method.

Photo editing. Another application is photo editing, in which we can remove object instances together with their associated shadows. Yi *et al.* [76] developed an image in-painting method for automatically removing specific objects by a given corresponding mask. With the results of instance segmentation methods, we can remove specific objects but leave shadows cast by the objects on the ground; see Figure 13 (c). With the help of our instance shadow detection results (Figure 13 (b)), we can remove the objects with their shadows altogether, as shown in Figure 13 (d).



Fig. 13: Instance shadow detection enables us to easily remove objects (e.g., dog and person) with their associated shadows altogether.

(a) Original image 1 (c) Naïve cut-and-paste

Fig. 14: We estimate the light direction and incorporate a virtual red post into each image with a simulated shadow, following [75].

Further, we can efficiently transfer an object together with its shadow from one photo to another. Figure 15 shows an example, in which we remove the girl together with her shadow from (b) and paste them together onto (a) in a smaller size. Clearly, if we simply paste the girl and shadow to (a), the shadow is not consistent with the real shadows in the target photo; see (c). Thanks to instance shadow detection, which outputs individual masks for objects together with their associated shadow instances, as well as the estimated 2D light direction. So, we can achieve light-aware photo editing by using the estimated light directions in both photos to adjust the shadow images when transferring the girl object from one photo to the other; see (d).

7 CONCLUSION

This paper presents instance shadow detection, targeting to predict shadow instances, object instances, and their relations. To

(b) Original image 2

(d) Light-aware shadow

Fig. 15: When we cut-and-paste objects from one photo to the other, instance shadow detection results enable us not only to extract object and shadow instances together but also to adjust the shadow shape according to the estimated light direction.

approach this task, we first prepare a new dataset and a new evaluation metric. Our dataset contains 1,100 images with labeled masks of 4,262 pairs of shadow instances, object instances, and shadowobject associations, while the evaluation metric promotes quantitative evaluation of instance shadow detection performance. We also design a new single-stage fully-convolutional network for instance shadow detection by directly learning the relation between shadow instances and object instances in an end-to-end manner. Further, we propose the bidirectional relation learning module, the deformable maskIoU head, and the shadow-aware copy-andpaste augmentation to improve the detection performance. Finally, we show the superiority of our method on the benchmark dataset and demonstrate the applicability of our method on light direction estimation and photo editing.

In the future, we plan to improve the performance of instance shadow detection by exploring the knowledge from the existing data prepared for other relevant vision tasks, *e.g.*, shadow detection and instance segmentation, from synthetic data generated by computer graphic techniques and from unlabeled data downloaded from the Internet. Also, we plan to explore more applications based on the shadow-object association results.

ACKNOWLEDGMENTS

This work was supported by the project MMT-p2-21 of the Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong, and the Hong Kong Research Grants Council under General Research Fund (CUHK 14201620 & CUHK 14201321).

REFERENCES

- T. Wang*, X. Hu*, Q. Wang, P.-A. Heng, and C.-W. Fu, "Instance shadow detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1880–1889, *Joint first authors.
- [2] T. Wang*, X. Hu*, C.-W. Fu, and P.-A. Heng, "Single-stage instance shadow detection with bidirectional relation learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1–11, *Joint first authors, oral presentation.
- [3] L. Hou, T. F. Y. Vicente, M. Hoai, and D. Samaras, "Large scale shadow annotation and detection using lazy annotation and stacked CNNs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 4, pp. 1337–1351, 2021.
- [4] X. Hu, T. Wang, C.-W. Fu, Y. Jiang, Q. Wang, and P.-A. Heng, "Revisiting shadow detection: A new benchmark dataset for complex world," *IEEE Transactions on Image Processing*, vol. 30, pp. 1925–1934, 2021.
- [5] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P.-A. Heng, "Direction-aware spatial context features for shadow detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7454–7462.
- [6] 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.
- [7] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Automatic feature learning for robust shadow detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1939–1946.
- [8] 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 *European Conference on Computer Vision*, 2018, pp. 662–678.
- [9] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Automatic shadow detection and removal from a single image," *IEEE Transactions* on Pattern Analysis and Machine Intelligence, vol. 38, no. 3, pp. 431– 446, 2016.
- [10] 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 *European Conference on Computer Vision*, 2016, pp. 816–832.
- [11] Q. Zheng, X. Qiao, Y. Cao, and R. W. Lau, "Distraction-aware shadow detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5167–5176.
- [12] 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 *European Conference on Computer Vision*, 2018, pp. 121–136.
- [13] Z. Chen, L. Zhu, L. Wan, S. Wang, W. Feng, and P.-A. Heng, "A multi-task mean teacher for semi-supervised shadow detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5611–5620.
- [14] H. Hao, S. Baireddy, E. Bartusiak, M. Gupta, K. LaTourette, L. Konz, M. Chan, M. L. Comer, and E. J. Delp, "Building height estimation via satellite metadata and shadow instance detection," in *Automatic Target Recognition XXXI*, vol. 11729, 2021, pp. 175 – 190.
- [15] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask Scoring R-CNN," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6409–6418.
- [16] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets V2: More deformable, better results," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9308–9316.

- [17] E. Salvador, A. Cavallaro, and T. Ebrahimi, "Cast shadow segmentation using invariant color features," *Computer Vision and Image Understanding*, vol. 95, no. 2, pp. 238–259, 2004.
- [18] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios, "Illumination estimation and cast shadow detection through a higher-order graphical model," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 673–680.
- [19] J. Tian, X. Qi, L. Qu, and Y. Tang, "New spectrum ratio properties and features for shadow detection," *Pattern Recognition*, vol. 51, pp. 85–96, 2016.
- [20] J. Zhu, K. G. Samuel, S. Z. Masood, and M. F. Tappen, "Learning to recognize shadows in monochromatic natural images," in *IEEE Conference* on Computer Vision and Pattern Recognition, 2010, pp. 223–230.
- [21] T. F. Y. Vicente, M. Hoai, and D. Samaras, "Leave-one-out kernel optimization for shadow detection," in *IEEE International Conference* on Computer Vision, 2015, pp. 3388–3396.
- [22] R. Guo, Q. Dai, and D. Hoiem, "Single-image shadow detection and removal using paired regions," in *IEEE Conference on Computer Vision* and Pattern Recognition, 2011, pp. 2033–2040.
- [23] T. F. Y. Vicente, M. Hoai, and D. Samaras, "Leave-one-out kernel optimization for shadow detection and removal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 682–695, 2018.
- [24] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Detecting ground shadows in outdoor consumer photographs," in *European Conference on Computer Vision*, 2010, pp. 322–335.
- [25] X. Huang, G. Hua, J. Tumblin, and L. Williams, "What characterizes a shadow boundary under the sun and sky?" in *IEEE International Conference on Computer Vision*, 2011, pp. 898–905.
- [26] L. Shen, T. Wee Chua, and K. Leman, "Shadow optimization from structured deep edge detection," in *IEEE Conference on Computer Vision* and Pattern Recognition, 2015, pp. 2067–2074.
- [27] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, and D. Samaras, "Shadow detection with conditional generative adversarial networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 4510–4518.
- [28] J. Wang, X. Li, and J. Yang, "Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1788–1797.
- [29] B. Ding, C. Long, L. Zhang, and C. Xiao, "ARGAN: Attentive recurrent generative adversarial network for shadow detection and removal," in *IEEE International Conference on Computer Vision*, 2019, pp. 10213– 10222.
- [30] Z. Chen, L. Wan, L. Zhu, J. Shen, H. Fu, W. Liu, and J. Qin, "Triplecooperative video shadow detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2715–2724.
- [31] L. Qu, J. Tian, S. He, Y. Tang, and R. W. Lau, "DeshadowNet: A multi-context embedding deep network for shadow removal," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4067–4075.
- [32] X. Hu, Y. Jiang, C.-W. Fu, and P.-A. Heng, "Mask-ShadowGAN: Learning to remove shadows from unpaired data," in *IEEE International Conference on Computer Vision*, 2019, pp. 2472–2481.
- [33] H. Le and D. Samaras, "Shadow removal via shadow image decomposition," in *IEEE International Conference on Computer Vision*, 2019, pp. 8578–8587.
- [34] X. Cun, C. Pun, and C. Shi, "Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting GAN," in AAAI Conference on Artificial Intelligence, 2020, pp. 10680–10687.
- [35] L. Zhang, C. Long, X. Zhang, and C. Xiao, "RIS-GAN: explore residual and illumination with generative adversarial networks for shadow removal," in AAAI Conference on Artificial Intelligence, 2020, pp. 12829– 12836.
- [36] H. Le and D. Samaras, "From shadow segmentation to shadow removal," in European Conference on Computer Vision, 2020, pp. 264–281.
- [37] Z. Liu, H. Yin, Y. Mi, M. Pu, and S. Wang, "Shadow removal by a lightness-guided network with training on unpaired data," *IEEE Trans*actions on Image Processing, vol. 30, pp. 1853–1865, 2021.
- [38] L. Fu, C. Zhou, Q. Guo, F. J. Xu, H. Yu, W. Feng, Y. Liu, and S. Wang, "Auto-exposure fusion for single-image shadow removal," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10571–10580.
- [39] Z. Liu, H. Yin, X. Wu, Z. Wu, Y. Mi, and S. Wang, "From shadow generation to shadow removal," in *IEEE Conference on Computer Vision* and Pattern Recognition, 2021, pp. 4927–4936.

- [40] Y.-H. Lin, W.-C. Chen, and Y.-Y. Chuang, "BEDSR-Net: A deep shadow removal network from a single document image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 905–12 914.
- [41] D. Liu, C. Long, H. Zhang, H. Yu, X. Dong, and C. Xiao, "ARShadow-GAN: Shadow generative adversarial network for augmented reality in single light scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8139–8148.
- [42] Y. Hong, L. Niu, and J. Zhang, "Shadow generation for composite image in real-world scenes," in AAAI, 2022.
- [43] X. C. Zhang, J. T. Barron, Y. Tsai, R. Pandey, X. Zhang, R. Ng, and D. E. Jacobs, "Portrait shadow manipulation," ACM Transactions on Graphics (SIGGRAPH), vol. 39, no. 4, p. 78, 2020.
- [44] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *IEEE Conference on Computer Vision* and Pattern Recognition, 2016, pp. 3150–3158.
- [45] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Conference on Neural Information Processing Systems*, 2015, pp. 1990–1998.
- [46] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*, 2016, pp. 75–91.
- [47] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instanceaware semantic segmentation," in *IEEE Conference on Computer Vision* and Pattern Recognition, 2017, pp. 2359–2367.
- [48] Z. Hayder, X. He, and M. Salzmann, "Boundary-aware instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5696–5704.
- [49] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "MaskLab: Instance segmentation by refining object detection with semantic and direction features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4013–4022.
- [50] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.
- [51] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.
- [52] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun, "MegDet: A large mini-batch object detector," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6181–6189.
- [53] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.
- [54] X. Chen, R. Girshick, K. He, and P. Dollár, "TensorMask: A foundation for dense object segmentation," in *IEEE International Conference on Computer Vision*, 2019.
- [55] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang, "SSAP: Single-shot instance segmentation with affinity pyramid," in *IEEE International Conference on Computer Vision*, 2019, pp. 642–651.
- [56] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "SOLO: Segmenting objects by locations," in *European Conference on Computer Vision*, 2020, pp. 649–665.
- [57] H. Ying, Z. Huang, S. Liu, T. Shao, and K. Zhou, "EmbedMask: Embedding coupling for one-stage instance segmentation," *arXiv preprint* arXiv:1912.01954, 2019.
- [58] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "SOLOv2: Dynamic and fast instance segmentation," in *Conference on Neural Information Processing Systems*, 2020.
- [59] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 906–13 915.
- [60] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *European Conference on Computer Vision*, 2020, pp. 282–298.
- [61] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 633–641.
- [62] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, 2019.
- [63] T. F. Y. Vicente, M. Hoai, and D. Samaras, "Noisy label recovery for shadow detection in unfamiliar domains," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3783–3792.
- [64] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in

context," in European Conference on Computer Vision, 2014, pp. 740-755.

- [65] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.
- [66] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: A simple and strong anchor-free object detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
- [67] R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
- [68] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.
- [69] A. Rosenfeld and J. Pfaltz, "Distance functions on digital pictures," *Pattern Recognition*, vol. 1, no. 1, pp. 33–61, 1968.
- [70] Z. Tian, H. Chen, X. Wang, Y. Liu, and C. Shen, "AdelaiDet: A toolbox for instance-level recognition tasks," https://git.io/adelaidet, 2019.
- [71] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.
- [72] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10781–10790.
- [73] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [74] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," arXiv preprint arXiv:1706.02677, 2017.
- [75] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Estimating natural illumination from a single outdoor image," in *ICCV*, 2009, pp. 183–190.
- [76] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *IEEE Conference* on Computer Vision and Pattern Recognition, 2020, pp. 7508–7517.