# Large scale shadow annotation and detection using lazy annotation and stacked CNNs

Le Hou, Tomás F. Yago Vicente, Minh Hoai, and Dimitris Samaras

**Abstract**—Recent shadow detection algorithms have shown initial success on small datasets of images from specific domains. However, shadow detection on broader image domains is still challenging due to the lack of annotated training data, caused by the intense manual labor required for annotating shadow data. In this paper we propose "lazy annotation", an efficient annotation method where an annotator only needs to mark the important shadow areas and some non-shadow areas. This yields data with noisy labels that are not yet useful for training a shadow detector. We address the problem of label noise by jointly learning a shadow region classifier and recovering the labels in the training set. We consider the training labels as unknowns and formulate label recovery as the minimization of the sum of squared leave-one-out errors of a Least Squares SVM, which can be efficiently optimized. Experimental results show that a classifier trained with recovered labels achieves comparable performance to a classifier trained on the properly annotated data. These results motivated us to collect a new dataset that is 20 times larger than existing datasets and contains a large variety of scenes and image types. Naturally, such a large dataset is appropriate for training deep learning methods. Thus, we propose a stacked Convolutional Neural Network architecture that efficiently trains on patch level shadow examples while incorporating image level semantic information. This means that the detected shadow patches are refined based on image semantics. Our proposed pipeline, trained on recovered labels, performs at state-of-the art level. Furthermore, the proposed model performs exceptionally well on a cross dataset task, proving the generalization power of the proposed architecture and dataset.

## 1 Introduction

Shadows are ubiquitous in images of natural scenes. On one hand, shadows provide useful cues about the scene including object shapes [43], light sources and illumination conditions [34, 46, 47], camera parameters and geo-location [30], and scene geometry [32]. On the other hand, the presence of shadows in images creates difficulties for many computer vision tasks from image segmentation to object detection and tracking. In all cases, detecting shadows, and subsequently removing them or reasoning about their shapes and sizes would be beneficial. Moreover, shadow-free images are of great interest for image editing, computational photography, and augmented reality, and the first crucial step is shadow detection.

The problem of single image shadow detection has been widely studied. Early work such as the illumination invariant approaches [15, 16] are based on physical modeling of the illumination and shadowing phenomena [46, 47]. These physics-based methods only work well with high quality images. In contrast, statistical learning approaches (e.g., [21, 23, 53, 59, 74]) have shown significant success in detecting shadows in consumer-grade photos and web quality images. The performance of these methods, however, depends on the quality and quantity of training images. Zhu *et al*. [74] and Guo *et al*. [21] were the first to collect sizable datasets of images with annotated shadows, the UCF and UIUC datasets respectively. These publicly available datasets with pixel-level annotations have led to important advances in the field. They enabled both systematic quantitative and qualitative evaluation of detection performance, as opposed to the prior practice of qualitative evaluation on a few selected images. In the past few years, several novel shadow detection methods (e.g., [23, 59]), gradually advanced state-of-the-art performance in these datasets, to the point of saturation. However, shadow detection is still far from being solved. Due to limited size, UIUC is biased by certain types of images such as objects in close range shots, whereas UCF is biased towards scenes with darker shadows. Because of their limited generality, cross-dataset performance (e.g., training on UIUC and testing on UCF) degrades significantly [22, 60]. The more recent ISTD dataset [64] contains 1870 shadow, shadow free, and shadow mask triplets. The procedure for collecting the ISTD and UIUC datasets was to take two pictures of the same scene once with an object casting a shadow, and once with the object removed. This method can only capture shadows in controlled environments. Many shadow types such as self shadows (e.g., shadows on the side of a building that does not "see" the light source) and shadows caused by unmovable objects cannot be captured. In order to facilitate the development of robust classifiers, a much larger and more general dataset is needed. However, creating a large shadow dataset would require enormous amount of effort, primarily for obtaining pixel-level annotations.

An alternative approach for robust shadow detection in unseen domains, is to utilize human input (or human aid) on particular input shadow images. Existing approaches [18, 19, 70] focus on shadow removal. Detecting shadow regions is considered a preprocessing step which relies on human input strokes on shadow and non-shadow regions. However, the shadow detection performance of these approaches has not been reported. The shadow detection results are not good enough, and are not intended to serve, as labels for large scale shadow detection training.

In this work we propose *"lazy annotation" with noisy label recovery*, a method that allows a human to quickly annotate shadow images with a few brush strokes [20]. Our method has several advantages compared to other existing approaches [18, 19, 36, 70]. First, our method recovers training labels when learning the shadow region detection model jointly, using the robust Least-Squares Support Vector Machines (LSSVM) [61]. This process is

• *Le Hou, Minh Hoai, and Dimitris Samaras are with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794. Tomás F. Yago Vicente is at A9.com: 130 Lytton Ave, Palo Alto, CA 94301. E-mail: lehhou@cs.stonybrook.edu*

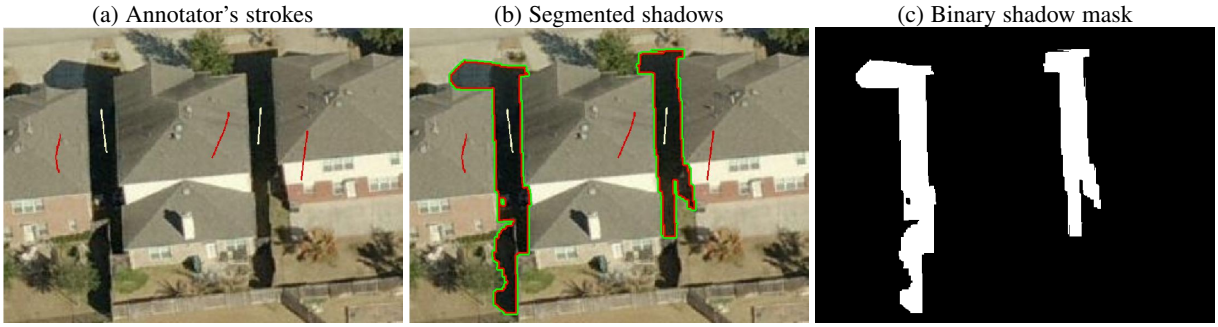



**Fig. 1: Lazy annotation pipeline for efficient labeling of shadow images.** a) An annotator is asked to draw some strokes on some (not all) shadow areas (white strokes) and non-shadow areas (red strokes). b) Automatically segmented shadow regions. c) Obtained shadow mask, mostly good with a few exceptions where some shadow regions are mis-labeled as non-shadow. Subsequently, the noisy labels are corrected using the label recovery method proposed in this paper.

applied on image clusters, instead of individual images separately. Thus, our method is more robust to noise in training labels introduced by rough brush strokes, compared to to existing approaches [18, 19, 36, 70]. Second, we extensively evaluate our interactive shadow labeling approach and for the first time, claim that the resulting shadow masks are as useful as carefully annotated ground truth masks, in terms of training shadow detection methods. Fig. 1 shows an example of this process, from the annotator's strokes to the generated binary shadow mask. Notice that the initial annotation is imperfect. Due to the nature of the task, label noise is asymmetric. The negative class (non-shadow) contains "dirty negatives", corresponding to missed shadows. The positive class (shadow) is significantly cleaner and more reliable, because the annotator is asked to label some shadows, so the shadow regions obtained are generally accurately labeled.

Our method jointly learns a classifier and recovers the training labels. In particular, we use LSSVM [61], consider the training labels as unknowns and formulate the problem as the minimization of the leave-one-out error. This leads to a binary quadratic programming problem where we constrain the fraction of originally labeled positive and negative instances that are flipped. To validate our approach, we relabeled the UIUC and UCF training sets using "lazy annotation". Experimental results show that a classifier trained with recovered labels achieves comparable performance to a classifier trained on the original, properly annotated datasets. Our label recovery method improves the accuracy of classifiers trained on "lazy" labels significantly. Furthermore, we show experimentally that label recovery is robust up to significant levels of label noise in the training set. To increase scalability, and since the leave-one-out error is most meaningful for similar data instances, we group similar images into smaller clusters and perform label recovery for each cluster independently. This leads to a scalable large-scale noisy label recovery algorithm, which combined with "lazy annotation" is the **first contribution** of this work.

To address the need for a large-scale shadow dataset, we collected the largest shadow dataset to date. This is the **second contribution** of this paper. Our dataset of almost 5000 images covers a wide range of scenes and is 20 times bigger than UCF [74], bringing shadow detection solutions under the large-data paradigm, and increasing the utility of deep learning approaches. The new SBU dataset is available at www3.cs.stonybrook.edu/~cvl/dataset.html. We carefully annotated shadow masks for 700 images, as a new benchmark for shadow detection. For the training set, images were first quickly labeled using "lazy annotation", then we run our label recovery method to clean up the annotations. Our dataset has already been found to be very useful for training shadow detectors in recent research by several groups [27, 35, 41, 64, 65].

As a **third contribution**, we propose a novel stacked Convolutional Neural Network (CNN) based approach for structured shadow prediction that takes advantage of the copious cleaned-up data. Given a large dataset, we design our architecture to learn not only local shadow cues, but also the discriminative global context. To do so, our semantics-aware stacked CNN architecture combines an Image-Level shadow predictor Network (ILN) and a patch-based CNN (Patch-CNN). The ILN can be any pretrained semantic segmentation network e.g., FCN [38], DeconvNet [42], FCN+CRF-RNN [72], SegNet [2]. In our case, we use a DeconvNet [42] model pretrained on PASCAL VOC 2012 for semantic segmentation, and fine tune it on shadow data. We use the outputs of an ILN together with the corresponding input RGB images to train the Patch-CNN with a random initialization. Thus, in the resulting Stacked-DeconvNet, the output of the ILN functions as an image-level shadow prior that is further refined by the more local appearance focus of the Patch-CNN.

We show that models trained on the newly collected SBU training set generalize better on the UCF test set (over 22% error reduction), compared to state-of-the-art methods [59, 61] trained on the UCF training set. This is remarkable as our training set does not overlap with the UCF dataset, proving the generalization ability of the trained model and the usefulness of the proposed dataset. Moreover, the proposed stacked architecture is a general framework in which any semantic segmentation network can be used as the ILN component for generating the image-level shadow prior. We have experimented with different semantic segmentation networks including [2, 38, 72], and in all cases, the final stacked architecture effectively refines the shadow prior, significantly improving the overall shadow detection performance. These experiments confirm the generality and robustness of the proposed architecture.

## 2 Previous Work

### 2.1 Shadow datasets and annotation

Annotated shadow datasets fostered work on shadow detection. However, there are only a few shadow datasets due to the cumbersome nature of annotation. The human annotator has to identify all the shadows in the image, and then properly delineate each shadow contour. It takes much time and attention for the many mouse clicks to create a polyline for each shadow. Free drawing to trace a shadow contour also takes considerable effort.

Wang *et al*. [64] and Guo *et al*. [21] generated a shadow annotation mask by taking two photographs of the same scene: a photo is taken with an occluder blocking the light source and casting a shadow in the scene, then a photo is taken when the occluder is removed. The shadow mask is generated by comparing the two images. Alternatively, they would take a second photo blocking the direct light source. The first approach is only applicable when the occluder is out of view and removable, whereas the second approach is limited to indoor environments with sufficient ambient light. Physically setting up the scene and taking the two shots is cumbersome, and this approach is not applicable to many scenes and shadow types. Qu *et al*. [50] and Gong *et al*. [18, 19] collected the Deshadow and the Shadow Removal datasets respectively, also using the method of taking two photographs of the same scene. However, the training dataset for the Deshadow paper has not been released, and no shadow mask ground truth has been given for either of these two datasets.

Existing publicly available shadow detection datasets are small or limited to cast shadows of movable objects only. Guo *et al*. [21, 22] report the cross-dataset performance of their model for UIUC and UCF datasets. A model trained on UCF training set performs well on the UCF test set, but not on the UIUC test set (90.0% versus 86.4% accuracy), and a model for UIUC dataset has much less accuracy when tested on the UCF test set (88.3% versus 79.4%).

### 2.2 Shadow Detection

A number of shadow detection methods have been developed in recent years. Guo *et al*. [21] proposed to model long-range interaction between pairs of regions of the same material, with two types of pairwise classifiers: same illumination condition and different illumination condition. Then, they combined the pairwise classifier and a shadow region classifier with a CRF. Similarly, Vicente *et al*. [63] proposed an MRF that combines a unary region classifier with a pairwise classifier and a shadow boundary classifier. These approaches achieved good shadow detection results, but required expensive ground-truth annotation. Khan *et al*. [23] were the first to use deep learning for shadow detection, achieving state-of-the-art results at the time. Vicente *et al*. [59] optimized a multi-kernel model for shadow based on leave-one-out estimates, obtaining even better shadow predictions than [23]. More recently, Shen *et al*. [53] proposed a CNN for structured shadow edge prediction. Nguyen *et al*. [41] proposed an approach based on a conditional GAN which was capable of changing its shadow sensitivity. Qu *et al*. [50] proposed a method for generating a 3-channel shadow matte. Wang *et al*. [64] proposed a method for joint shadow detection and removal. Very recent works [27, 35, 62, 65] are also based on deep learning, and significant advances have been made. Many of these advances were only possible because of the availability of the large-scale dataset described in this paper. Furthermore, the proposed stacked CNN architecture provides complementary benefits to the other deep learning methods and can be easily combined with them.

### 2.3 Noisy label recovery

The presence of label noise, also known as class noise, may severely degrade classification performance [17, 76]. Numerous methods seek robustness to noisy labels [12, 33, 40, 55]. For instance, Stempfel *et al*. [56] deal with training a binary Support Vector Machine (SVM) when the probability of flipping a label is constant and only depends on the true class. For this, they replace the objective functional by a uniform estimate of the corresponding noise-free SVM objective. This becomes a non-convex problem that can be solved with Quasi-Newton BFGS. Biggio *et al*. [5] compensate noise in the labels by modifying the SVM kernel matrix with a structured matrix modeling the noise. This approach only models random flips with fixed probability per class and adversarial flips. That is, for a set number of labels to be flipped, the adversary tries to maximize the classification error. These methods are designed to not be affected by label noise rather than to be effective in using noisy labels for training. Moreover, these methods focus on asymptotic behavior with unlimited training data. In contrast, as the training data is very limited for the shadow detection problem, we aim to make effective use of noisy labels. Furthermore, our method obviates the need of assumptions on the nature of the noise such as constant[55], class-dependency with fixed probability [5], limited noise ratios [40].

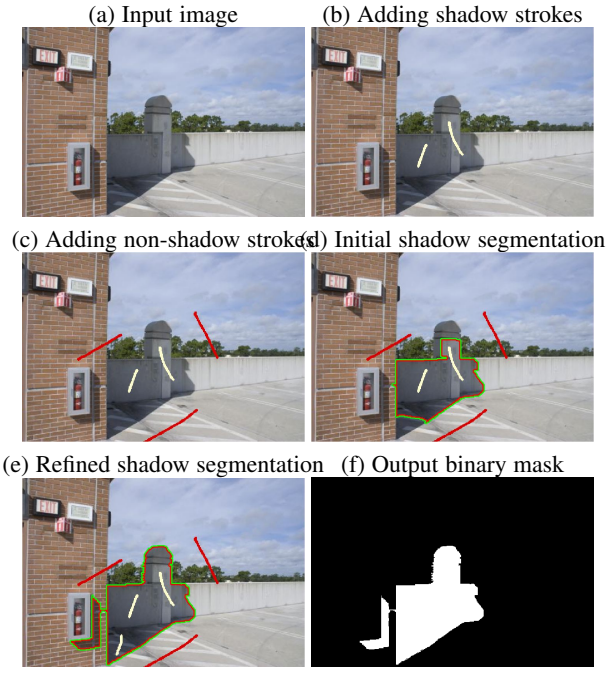



**Fig. 2: Lazy annotation pipeline.** a) Input image. b) Annotator's shadow strokes in white. c) Annotator's non-shadow strokes in red. d) Initial shadow segmentation in green (outer side) and red (inner side). e) Refined shadow segmentation with a final shadow stroke in the lower center of the image. f) Resulting binary mask.

## 3 Lazy Annotation

Our objective is to obtain ground truth shadow annotation with minimal effort and time. Generating good annotation typically requires manually segmenting all the shadows in an image. We simplify the annotation task by redefining its goal. Rather than segmenting all shadows, we instruct the annotator to focus on at least one shadow area of the image. This typically corresponds to the most prominent shadow. We use a semi-automatic shadow segmentation scheme requiring minimal annotator input. The annotator only has to draw a few strokes on shadow areas and a few additional strokes on non-shadow areas.

### 3.1 Lazy annotation pipeline

We illustrate our lazy annotation pipeline with an example image in Fig. 2. First, the annotator draws a few strokes (2-3) on areas of the image she considers relevant shadows, see Fig. 2.b. Then, the annotator draws a few strokes (2-3) on non-shadow areas surrounding the shadow, see Fig. 2.c. After that, a shadow segmentation based on these strokes is presented to the annotator, see Fig. 2.d. Then, the annotator is able to add a few additional strokes to refine the shadow segmentation interactively. In Fig. 2.e, the additional shadow stroke on the concrete ground grows the shadow region and even segments an extra shadow on the brick wall. The shadow mask resulting from the user annotation is depicted in Fig. 2.f. We interactively segment the images using the method of Gulshan *et al*. [20]. The method combines geodesic star convexity shape constraints with the Boykov-Jolly [6] energy formulation for image segmentation based on user strokes denoting foreground and background. In our case, shadows correspond to foreground. We modify the publicly available tool [20] to render a more streamlined user interface tailored to our task. Mouse interaction is only required for brush strokes. The remainder of the interface is commanded by keystrokes: Switching brush type (shadow or non-shadow stroke), advancing to refinement interactive stage, and signaling completion. Furthermore, a batch of images is loaded consecutively one after the next. With this tool, an annotator can typically label 3 images a minute on average.

### 3.2 Postprocessing

We frame shadow detection as a region classification problem. Hence, we need to generate region labels from the binary mask resulting from the lazy annotation. We followed the region segmentation process in [63] for shadow detection. First, we oversegment the image into SLIC [1] superpixels (see Fig. 3.a). Then, we apply Mean-shift clustering in Lab space and merge connected superpixels in the same cluster into a larger region, see Fig. 3.b.

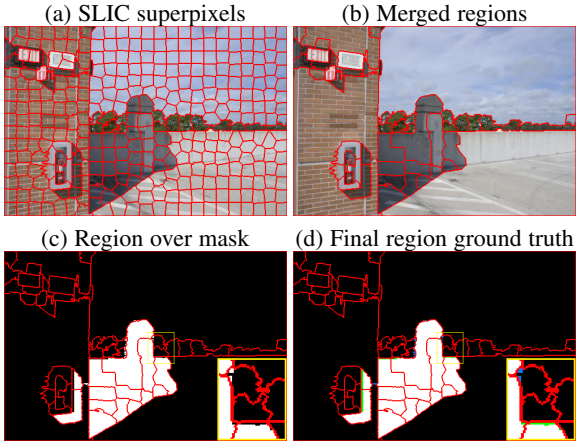



**Fig. 3: From lazy shadow mask to region labels.** a) Initial SLIC superpixels. b) Regions obtained by merging superpixels. c) Lazy mask overlaid on regions. d) Final region ground-truth.

We overlay the binary mask on the segmented regions (Fig. 3.c). If a region contains a majority of shadow pixels it is labeled positive, otherwise it is labeled negative. Overall, the proposed annotation approach is able to generate reasonably good region labels. Regions labeled as shadows are generally reliable whereas negatively labeled regions may contain missed shadows. For example in Fig. 3, a few small shadow regions on the brick wall in the top left corner of the image are labeled non-shadow.

## 4 Noisy Label Recovery

### 4.1 Formulation

In this section, we describe a method for noisy label recovery. We pose it as an optimization problem, where the labels of some training examples can be flipped to minimize the sum of squared leave-one-out errors. The basis of our formulation is that the leave-out-out error of kernel LSSVM is a linear function of the labels. Our noisy annotation recovery framework is based on the Least-Squares Support Vector Machine (LSSVM) [51, 57]. LSSVM has a closed-form solution, which, once computed, enables efficient finding of the solution for a reduced training set, by removing any one training data point. This permits reusing training data for further calibration, e.g., [24, 25, 66], and for noisy label recovery. We introduce a kernelized algorithm for noisy label recovery with non-linear kernels, which are important for shadow detection [59].

Given a training set of $n$ data points $\{\mathbf{x}_i\}_{i=1}^n$* and associated binary labels $\{y_i\}_{i=1}^n$, LSSVM optimizes the following:

$$\underset{\mathbf{w},b}{\text{minimize}}\ \lambda||\mathbf{w}||^2 + \sum_{i=1}^{n} s_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i)^2, \qquad (1)$$

where $s_i$ is the weight of the $i$-th instance. For high dimensional data (i.e., $\phi(\mathbf{x}_i)$ is large), it is more efficient to obtain the solution for $(\mathbf{w}, b)$ via the representer theorem, which states that $\mathbf{w}$ can be expressed as a linear combination of the training data, i.e., $\mathbf{w} = \sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i)$. Let $\mathbf{K}$ be the kernel matrix, $k_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. The objective function becomes:

$$\underset{\boldsymbol{\alpha},b}{\text{minimize}}\ \lambda\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} + \sum_{i=1}^{n} s_i(\mathbf{k}_i^T\boldsymbol{\alpha} + b - y_i)^2, \qquad (2)$$

where $\mathbf{k}_i$ is the $i^{th}$ column of matrix $\mathbf{K}$, and $s_i$ is the instance weight, allowing the assignment of different weights to different training instances.

Let $\overline{\boldsymbol{\alpha}} = [\boldsymbol{\alpha}, b], \overline{\mathbf{K}} = [\mathbf{K}; \mathbf{1}_n^T], \mathbf{R} = \begin{bmatrix} \lambda\mathbf{K} & \mathbf{0}_n \\ \mathbf{0}_n^T & 0 \end{bmatrix}$. Then Eq. (2) is equivalent to minimizing:

$$\lambda\overline{\boldsymbol{\alpha}}^T\mathbf{R}\overline{\boldsymbol{\alpha}} + \sum_{i=1}^{n} s_i(\bar{\mathbf{k}}_i^T\overline{\boldsymbol{\alpha}} - y_i)^2. \qquad (3)$$

This is an unconstrained quadratic program, and the optimal solution can be found by setting the gradient to zero, i.e., solving:

$$(\mathbf{R} + \overline{\mathbf{K}}diag(\mathbf{s})\overline{\mathbf{K}}^T)\overline{\boldsymbol{\alpha}} = \overline{\mathbf{K}}diag(\mathbf{s})\mathbf{y}, \qquad (4)$$

where $diag(\mathbf{s})$ is a matrix with the $i$-th entry in its main diagonal equals to $s_i$, and zero in all non-diagonal entires.

Let $\mathbf{C} = \mathbf{R} + \overline{\mathbf{K}}diag(\mathbf{s})\overline{\mathbf{K}}^T, \mathbf{d} = \overline{\mathbf{K}}diag(\mathbf{s})\mathbf{y}$. The solution for kernel LSSVM is: $\overline{\boldsymbol{\alpha}} = \mathbf{C}^{-1}\mathbf{d}$. Now suppose we remove the training data point $\mathbf{x}_i$, let $\mathbf{C}_{(i)}, \mathbf{d}_{(i)}, \overline{\boldsymbol{\alpha}}_{(i)}$ be the corresponding values when removing $\mathbf{x}_i$. We have $\overline{\boldsymbol{\alpha}}_{(i)} = \mathbf{C}_{(i)}^{-1}\mathbf{d}_{(i)}$. Note that, even though we remove $\mathbf{x}_i$ from the training data, we can still write $\mathbf{w}$ as the linear combination of $\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n)$ without excluding the term $\phi(\mathbf{x}_i)$. The matrices $\mathbf{K}, \overline{\mathbf{K}}, \mathbf{R}$ remain the same, and the only change is the removal of $s_i(\mathbf{k}_i^T\boldsymbol{\alpha} + b - y_i)^2$ from

*. Bold uppercase letters denote matrices (e.g., $\mathbf{K}$), bold lowercase letters denote column vectors (e.g., $\mathbf{k}$). $\mathbf{k}_i$ represents the $i^{th}$ column of the matrix $\mathbf{K}$. $k_{ij}$ denotes the scalar in the row $j^{th}$ and column $i^{th}$ of the matrix $\mathbf{K}$ and the $j^{th}$ element of the column vector $\mathbf{k}_i$. Non-bold letters represent scalar variables. $\mathbf{1}_n \in \Re^{n\times 1}$ is a column vector of ones, and $\mathbf{0}_n \in \Re^{n\times 1}$ is a column vector of zeros.

**Fig. 4: Examples of clusters of similar shadow images.** Clusters of images resulting from running modified PGP on SBU training set.

the objective function. Thus we have $\mathbf{C}_{(i)} = \mathbf{C} - s_i\overline{\mathbf{k}}_i\overline{\mathbf{k}}_i^T$ and $\mathbf{d}_{(i)} = \mathbf{d} - y_i s_i \overline{\mathbf{k}}_i$. Using the Sherman-Morrison formula, we have:

$$\mathbf{C}_{(i)}^{-1} = (\mathbf{C} - s_i\overline{\mathbf{k}}_i\overline{\mathbf{k}}_i^T)^{-1} = \mathbf{C}^{-1} + \frac{\mathbf{C}^{-1}s_i\overline{\mathbf{k}}_i\overline{\mathbf{k}}_i^T\mathbf{C}^{-1}}{1 - s_i\overline{\mathbf{k}}_i^T\mathbf{C}^{-1}\overline{\mathbf{k}}_i}, \quad (5)$$

Using the above equations to develop $\overline{\boldsymbol{\alpha}}_{(i)} = \mathbf{C}_{(i)}^{-1}\mathbf{d}_{(i)}$, and let $\mathbf{M} = \mathbf{C}^{-1}\overline{\mathbf{K}}$ and $\mathbf{H} = \mathbf{M}^T\overline{\mathbf{K}}$, we obtain the following formula for the leave-one-out (LOO) weight vector:

$$\overline{\boldsymbol{\alpha}}_{(i)} = \overline{\boldsymbol{\alpha}} + \frac{(\overline{\boldsymbol{\alpha}}^T\overline{\mathbf{k}}_i - y_i)s_i}{1 - s_i h_{ii}}\mathbf{m}_i. \quad (6)$$

The LOO error can therefore be computed efficiently: $\overline{\boldsymbol{\alpha}}_{(i)}^T\overline{\mathbf{k}}_i - y_i = \frac{\overline{\boldsymbol{\alpha}}^T\overline{\mathbf{k}}_i - y_i}{1 - s_i h_{ii}}$. Substituting $\overline{\boldsymbol{\alpha}} = \mathbf{M}diag(\mathbf{s})\mathbf{y}$ into the above, the leave-one-out error becomes:

$$\frac{\overline{\mathbf{k}}_i^T\mathbf{M}diag(\mathbf{s})\mathbf{y} - y_i}{1 - s_i h_{ii}}. \quad (7)$$

Let $\mathbf{P} = diag(\mathbf{s})\mathbf{H}$ and recall that $\mathbf{H} = \mathbf{M}^T\overline{\mathbf{K}}$. The leave-one-out error is: $\frac{\mathbf{p}_i^T\mathbf{y} - y_i}{1 - p_{ii}}$.t Let $\mathbf{e}_i$ be the $i^{th}$ column of the identity matrix of size $n$, and $\mathbf{a}_i = \frac{\mathbf{p}_i - \mathbf{e}_i}{1 - p_{ii}}$, then the leave-one-out error becomes $\mathbf{a}_i^T\mathbf{y}$. Because the vector $\mathbf{a}_i$ only depends on the data, the leave-one-out error is a linear function of the label vector $\mathbf{y}$.

Let $\mathcal{P}, \mathcal{N}$ be the indices of (noisy) positive and negative training instances respectively, i.e. $\mathcal{P} = \{i|y_i = 1\}$ and $\mathcal{N} = \{i|y_i = 0\}$. We pose noisy label recovery as the optimization problem that minimizes the sum of squared leave-one-out errors:

$$\underset{y_i \in \{0,1\}}{\text{minimize}} \sum_{i=1}^{n} (\mathbf{a}_i^T\mathbf{y})^2, \quad (8)$$

$$\text{s.t.} \sum_{i \in \mathcal{P}} y_i \geq \alpha|\mathcal{P}| \quad \text{and} \quad \sum_{i \in \mathcal{N}} y_i \leq (1 - \beta)|\mathcal{N}|. \quad (9)$$

In the above $|\mathcal{P}|, |\mathcal{N}|$ are the original number of positive and negative training instances respectively, and $\alpha, \beta$ are parameters of the formulation ($0 \leq \alpha, \beta \leq 1$). The constraint of the above optimization problem requires that the proportion of original positive training instances that remains positive must be greater than or equal to $\alpha$. It also limits the proportion of flipped negative data points to be at most $1 - \beta$. If $\alpha = \beta = 1$, none of the training labels can be flipped.

### 4.2 Large-scale Noisy Label Recovery

The label recovery method previously described requires solving a binary quadratic program in which the number of variables is the same as the number of image regions. This full-scale optimization problem is too big to be solved on a large dataset at once. To bypass this issue, we propose here a simple but effective approach. We divide images into clusters of similar images, and perform label recovery for each cluster independently. This approach is motivated by the fact that our label recovery algorithm is based on optimizing the leave-one-out errors. Perhaps the wrong label of a region can be corrected because the region is similar to other regions with correct labels. As such, for label recovery, dissimilar regions do not have much impact on each other. Hence, it makes sense to recover labels within clusters of similar images.

Using our approach, we can recover the labels of hundreds of thousands of image regions. This approach allows us to consider superpixels rather than larger regions. We oversegment images using Linear Spectral Clustering [73]. The oversegmentation minimizes frequent inaccuracies in shadow segmentation where small shadow areas "leak" into large non-shadow regions. After all shadows are a well known segmentation confounder.

For image-set clustering, we use a modified version of the Parametric Graph Partitioning method (PGP) [68], which works well for image and video segmentation [69]. PGP does not require setting the number of clusters, as opposed to $k$-means clustering.

**Image-set clustering details.** We aim to cluster images depicting similar scenes and therefore similar shadows (the appearance of shadows depends on scene properties, including illumination, color, and texture of materials). For feature representation, we use GIST [44], and the *a* and *b* components of the Lab color space. We compute histograms of *a* and *b* from the shadow areas and their surroundings. For this, we use the initial annotated shadow mask and dilate it with an area ratio of 3:2 (shadow vs non-shadow). We used a 30-bin histogram for the *a* and *b* features separately, and the original 512-bin histogram for GIST.

## 5 Stacked-CNN Architecture for Shadow Segmentation

PGP [68] groups data into clusters by finding and removing between-cluster edges from a weighted graph, where the graph nodes are the data points and the edges define neighborhood relationships where the pair-wise similarity distances are the edge weights. Given the graph, a two-component Weibull Mixture Model is fitted over the edge weights. Then, we use the cross-point of the two Weibull components as the critical value that represents the cut-off between the within-cluster edge weights and the between-cluster edge weights. After the critical value is computed, the edges with weights higher than the critical value are identified as between-cluster edges and removed, with the subsequent disjoint sets of sub-graphs as the final clustering result. For the shadow image-set clustering problem, initial neighborhood relationships are not explicitly defined. Therefore, we construct the data graph by linking data nodes with their $k$-nearest neighbors. Each node represents an image. We use Earth Mover's Distance

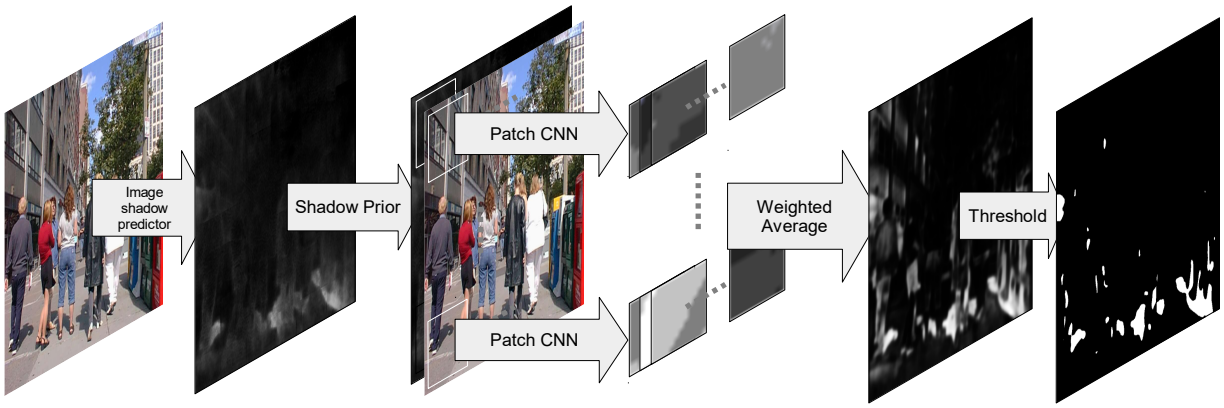



**Fig. 5: The proposed pipeline for shadow segmentation.** A conventional semantic image segmentation CNN [2, 38, 42, 72] takes an RGB image and outputs an image level shadow prior map. Then a patch level CNN examines local texture and color via a sliding window approach, taking an RGBP (P is the image level shadow Prior channel) image patch and outputs a local shadow prediction map. The probability of each pixel being a shadow pixel is computed by averaging results from different patches.

(EMD) as the distance metric for the $a$ and $b$ color histograms, and Euclidean ($L_2$) distance for the GIST features. Given the three similarity distances per node pair, we normalize the EMD and $L_2$ distance values to have zero mean and unit variance, perform PCA, and take the first principal component as the combined similarity distance for constructing the $k$-nearest neighbor data graph.

Once the clusters are computed by applying PGP on the graph, we add a post-processing step to enforce the size of each cluster to be within $n_{min} = 10$ to $n_{max} = 60$ images. We iteratively merge small clusters (with less than $n_{min}$ images) into the closest cluster; i.e, the cluster that has the member with the lowest combined similarity distance to a member of the small cluster. We re-apply PGP to the clusters larger than $n_{max}$ until the sizes of all resulting clusters fall within the desired range.

Most previous shadow detection methods [29, 53, 59, 60, 63] are based on classification of image regions using local color and texture cues. This approach, however, ignores global semantic information, which is useful for disambiguation. For example, without reasoning about global semantics, a dark cloud in the sky might be misclassified as a shadow region. On the other hand, CNN-based semantic segmentation methods [2, 9, 10, 38, 42, 48, 72] focus on whole image level semantic information. However, shadow regions have distinct texture and color compared to non-shadow regions, which must be examined in detail. In this section, we propose a stacked CNN architecture that trains a semantics-aware patch level CNN, a method that combines global semantics with local cues for shadow detection.

Our method is based on combining two neural networks. Combining multiple neural networks has been used in many applications [11, 13, 28, 31, 49, 52, 54]. One approach is to train multiple neural networks separately then combine their predictions [11, 28, 54]. Another approach is to combine the feature maps of neural networks instead of the final predictions [31]. These approaches, require the networks to share the same input/output structure and learning objective. Instead we propose to stack two CNNs into a single stream, as shown in Fig. 5. The two networks can have heterogeneous input/output representation and learning objectives.

In order to introduce global semantics, we first train an image-level semantic segmentation network, such as FCN [38], DeconvNet [42], FCN+CRF-RNN [72], SegNet [2], for shadow localization. Subsequently, the probability map predicted by the Image-Level shadow predictor Network (ILN) for a training image is attached to the original RGB image as an additional channel. We refer to the additional channel as the image-level shadow prior channel or P. Finally, we train a patch-based CNN on RGBP patches to predict local shadow pixels, referred to as Patch-CNN. The final prediction of a shadow pixel is a weighted average over the prediction outputs for all patches containing this pixel. The use of a Patch-CNN in addition to the image-level semantic segmentation network has a "resolution" advantage. Although, the deep layers of ILN can extract semantic information, the high resolution texture and color information is minimized due to max-pooling and down-sampling. Therefore, a local Patch-CNN is necessary to refine the segmentation result. The Patch-CNN learns from millions of training patches, leading to a more robust local shadow classifier. By including the output of the ILN as an image-level shadow prior channel in the input, we effectively incorporate semantic information into the Patch-CNN to generate improved shadow masks, see qualitative examples in Fig. 6.

### 5.1 Image-Level network (ILN) details

The proposed Stacked-CNN is a combination of an ILN with a Patch-CNN. In theory, any semantic segmentation network can be used as an ILN. In this paper, we evaluate several specific semantic segmentation networks as described below.

**Fully Convolutional Network (FCN).** We train an FCN [38] whose architecture is adopted from the VGG-16 network by changing fully connected layers to convolutional layers [38], on shadow images to generate the image level shadow prior. We adopt the FCN-8 network architecture and implementation [38] and fine-tune the FCN-8 that is already trained on the PASCAL VOC dataset [14], using the given shadow images and masks.

**CRF-RNN.** We use the same CRF-RNN layer implementation applied on the MS COCO dataset [37] by the authors of CRF-RNN [72] on top of the FCN: the segmentation output of the FCN is connected to the input of the CRF-RNN layer. Thus, the FCN+CRF-RNN outputs a more refined segmentation map compared to the FCN. We also initialize the FCN part of FCN+CRF-RNN with the VGG-16 network (as suggested by the FCN paper [38]), and initialize the CRF-RNN parameters randomly. The whole FCN+CRF-RNN network is trained end-to-end.

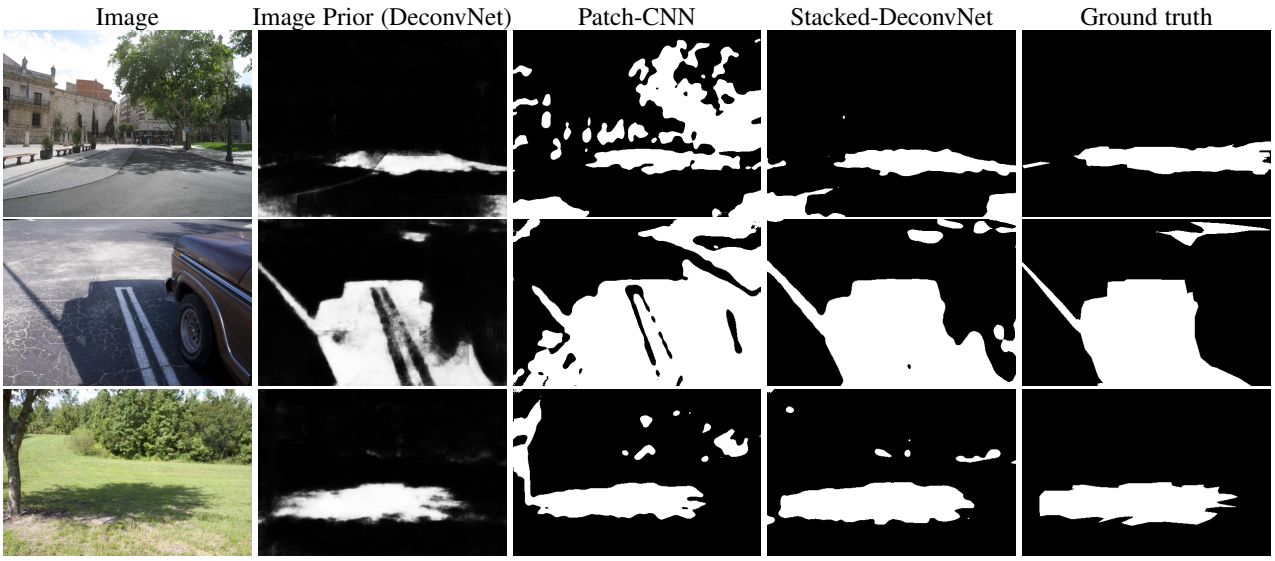



**Fig. 6: Shadow segmentation examples.** Qualitative results using Patch-CNN on RGB images, and on RGBP (P is the image level shadow prior) images (Stacked-DeconvNet). The Stacked-DeconvNet achieves the best results by incorporating both semantic and subtle local texture and color information. For example, in the first image, although the color and texture of the tree resemble a shadow, we can exclude the tree pixels thanks to the DeconvNet generated shadow prior.

**SegNet.** We adopt the same network architecture and implementation that the authors of SegNet [2] used on the CamVid dataset [7]. We use a SegNet initialized from a trained VGG-16 network, and fine-tune it on shadow datasets. The initialization method is also adopted from the SegNet implementation: the fully connected layers of VGG-16 are transformed into $1\times1$ convolutional layers.

**DeconvNet.** We adopt the network architecture and implementation that the authors of DeconvNet [42] used on the PASCAL VOC dataset [14]. We fine-tune the DeconvNet that is already trained on PASCAL VOC using the given shadow masks. Additionally, same to the DeconvNet approach [42], we also apply a Dense-CRF as a postprocessing step.

**ADNet.** We use the shadow detection network proposed by the authors of A+D Net [35]. We simply use the network trained by the authors, and the shadow detection results obtained by the authors.

We combine a patch-level CNN (denoted as Patch-CNN) with the aforementioned ILNs in a two-step fashion: we first train a ILN, get the ILN's prediction, then train a Patch-CNN using the RGB + P channels, as illustrated in Fig. 5. The resulting methods are **Stacked-FCN**, **Stacked-CRF-RNN**, **Stacked-SegNet**, **Stacked-DeconvNet**, and **Stacked-ADNet**, respectively.

### 5.2 Patch-CNN details

We build a patch level CNN with structured output for local shadow segmentation, as shown in Fig. 7. The loss function is the average negative log-likelihood (binary cross-entropy) of the prediction of every pixel. We extract image patches for training in three ways. 25% of the patches are extracted at random image locations to include patches of various textures and colors. 50% are extracted on Canny edges [8] to include hard-to-classify boundaries. 25% are extracted at shadow locations to guarantee a minimum percent of positive instances. This results in an overall balanced number of shadow pixels and non-shadow pixels in the training batches for stochastic gradient descent. During testing, we input all the overlapping patches of each image to the Patch-CNN. Thus every pixel has a maximum of $32 \times 32 = 1024$ predicted values from all the different patches. We use a weighted average to fuse the multiple predictions. More precisely, suppose there are $n$ patches containing the pixel, the distances between the pixel and the center of those patches are $d_1, d_2, \ldots, d_n$, and the predicted shadow probabilities are $p_1, p_2, \ldots, p_n$ respectively. Then the fused shadow probability is: $p = (\sum_i G(d_i;\sigma)p_i)/\sum_i G(d_i;\sigma)$, where $G(d_i;\sigma)$ is a zero-mean Gaussian with variance $\sigma^2$. In our experiments we use $\sigma^2 = 8$.

## 6 SBU Shadow Dataset

We have collected a new shadow dataset, one that is significantly larger and more diverse than the existing datasets [21, 74], and used lazy annotation to quickly annotate the images. Our dataset is available at: http://www3.cs.stonybrook.edu/~cvl/dataset.html

**Image collection.** To compile our dataset, we collected almost 5,000 images containing shadows. A quarter of the images came from the MS COCO dataset [37]. The rest were collected from the web. This image collection is significantly larger than the existing UCF [74] and UIUC [21] datasets, which contain less than 400 images combined. This image collection is also more diverse than existing datasets, which consist of images from a few specific domains (e.g., close shots of objects predominate in UIUC, whereas the majority of the UCF images are scenes with darker shadows and objects). The image collection covers a wide range of scenes including urban, beach, mountain, roads, parks, snow, animals, vehicles, and houses. It also contains different picture types including aerial, landscape, close range, and selfies. We split the images into two subsets for training and testing. The training subset contains about 85% of the images.

**Shadow image annotation.** We divided the image collection into disjoint training and test subsets and used two different approaches for annotation. For 700 test images, we carefully annotated the images, aiming for pixel accuracy to ensure the validity of numerical evaluation. We will refer to this test set as **SBU-Test**. For training images, we used *lazy labeling* to quickly annotate a large set of images. For lazy labeling, we drew a few strokes on shadow areas and a few other strokes on non-shadow areas. These strokes were

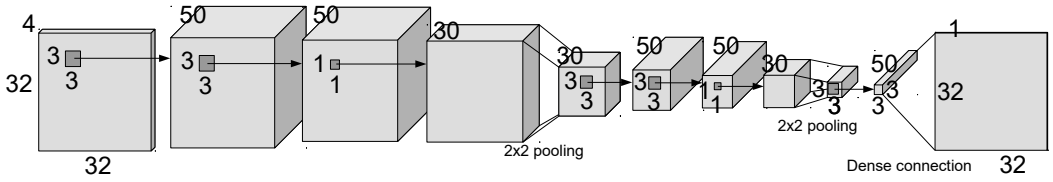



**Fig. 7: Patch-CNN with structured output.** The input is a $32 \times 32$ RGBP (RGB + image level shadow Prior) image, the output is a $32 \times 32$ shadow probability map.

used as shadow and non-shadow seeds for geodesic convexity image segmentation [20]. Fig. 1 illustrates this procedure. With lazy labeling, we were able to annotate the dataset quickly, at the rate of 3 to 4 images per minute. However, the obtained annotation was noisy. In particular, there were many "dirty negatives"—shadow regions that were incorrectly labeled as negative. This was due to misclassification of shadow regions or poor segmentation (image regions contain both shadow and non-shadow pixels). Dirty negatives were more prevalent than "dirty positives". Since we focused on drawing strokes on major shadow areas, the chosen shadow areas were generally well segmented. The final dataset contains images with shadow labels that have been "cleaned" using the method described in Section 4.1. Hereafter, we refer to the dataset with noisy labels as **SBU-Train-Noisy** and the dataset with recovered labels as **SBU-Train-Recovered**.

## 7 Label Recovery Evaluation

In this section we show that our label recovery approach is effective and yields better shadow region detection performance. In all experiments, we use an Lease-Squares SVM (LSSVM) with a $\mathcal{X}^2$ kernel for label recovery on each image cluster obtained with the proposed PGP-based clustering (except for the non-shadow datasets used in Sec. 7.4 where we do not apply clustering). We use an LSSVM with linear kernel as a shadow detector/classifier, to evaluate the quality of the recovered training set. The LSSVM with linear kernel is not the state-of-the-art method for shadow region detection but generates reasonable results[59]. For completeness, we also include the effect of noisy label recovery on CNN training.

### 7.1 Relabeling UCF and UIUC with lazy annotation

We relabeled the original UCF and UIUC training sets using lazy annotation, which we will refer to as UCF-Lazy and UIUC-Lazy, respectively. Lazy annotation takes roughly 20 seconds per image (average measured over 110 UCF images). In contrast, conventional shadow annotation with polylines takes 4.7 minutes per image (average measured over the first 30 UCF images).

We train a region classifier using the lazy labels and measure classification performance in the respective test sets. We now evaluate shadow detection in terms of average precision (AP), as we are more interested in evaluating the relative impact on classification performance of the different label types without thresholding. Since each data point corresponds to a region, we weigh each region by its area in pixels to approximate pixel AP.

Table 1 shows the classification performance in terms of AP. For UIUC, the testing performance of the model trained on lazy labels deteriorates by over 10% compared to training with the original labels (79.5% vs 88.5%). For UCF, training with lazy labels is only slightly worse than training with the original labels, 73.5% versus 74.5%. We then run the proposed label recovery method on the lazily annotated datasets. As shown in Table 1,

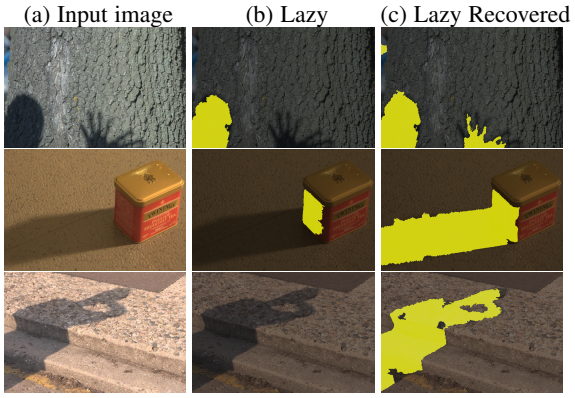



**Fig. 8: Shadow detection comparison between models trained with lazy labels and recovered labels on UIUC.** a) Input image. b) Detection results from model trained on lazy labels overlaid in yellow. c) Detection results from model trained on recovered lazy labels overlaid in yellow.

**TABLE 1: Classification performance on UIUC and UCF test sets.** Comparison of AP achieved by a linear LSSVM region classifier trained with original carefully annotated labels (Original), lazily annotated labels (Lazy), and recovered lazy labels (Recovered).

| Train data | Test data | AP |
|---|---|---|
| UIUC-Original | UIUC | 88.5 |
| UIUC-Lazy | UIUC | 79.5 |
| UIUC-Recovered | UIUC | 87.2 |
| UCF-Original | UCF | 74.5 |
| UCF-Lazy | UCF | 73.5 |
| UCF-Recovered | UCF | 75.6 |

for UIUC the same classifier trained on recovered labels (UIUC-Recovered) achieves comparable testing performance to training with the original ground-truth labels: 87.2% vs 88.5%. Label recovery achieves almost 10% improvement compared to training with lazy labels. Fig. 8 shows qualitative comparisons of shadow detection for the region classifier trained on UIUC-Lazy and UIUC-Recovered. In the depicted examples. notice how the model trained on recovered labels correctly detects more shadow regions.

Interestingly, for UCF label recovery improves the testing performance to 75.6%, outperforming the model trained on the original labels with 74.5% AP. This indicates that some of the ground-truth labels provide pernicious training examples.This may not necessarily imply that the labels were wrongly annotated.

With lazy annotation, the annotator often misses less prominent shadows, and shadows in the background areas. For instance, in Fig. 9 we show an example from the UCF training set: The shadow of the smaller brown column in the top right of the image is missed by the annotator. The initial shadow mask from lazy annotation is overlaid in blue. However, the label recovery method is able to correct the annotation; in Fig. 3.(c) the recovered shadow

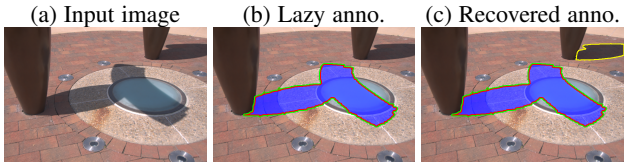



**Fig. 9: Example of label recovery.** a) Input image; b) Lazy annotation shadow mask overlaid in blue, outer contour in green, inner contour in red; c) Recovered regions with flipped shadow label are shown with yellow contours.

is depicted with a yellow contour.

Overall, these experiments suggest that we can achieve similar results with recovered lazy labels as with carefully annotated labels, but with significantly smaller annotation effort.

### 7.2 Comparing to existing interactive shadow detection methods

In this section we compare our "lazy annotation" with noisy label recovery approach, to existing baseline methods [18, 19, 36]. We compare to the **interactive shadow labeling method** [18, 19] and the **closed-form matting method** [36] using the same user stroke input. Our method gives shadow detection results on three different phases during the pipeline:

1) **Noisy labels**: shadow prediction results using user stroke input. Noisy labels are obtained per image, detailed in Section 3 in the main submission.
2) **Recovered labels**: recovered, relatively clean labels, by learning from all noisy labels across all images. This is detailed in Section 4 in the main submission.
3) **Prediction given by Stacked-CNN**: our final, fully automatic shadow prediction results. The Stacked-CNN model is trained on recovered labels. In particular, we use Stacked-DeconvNet as our Stacked-CNN method, see Sec. 8.1.

We compare all methods on the UCF test set [74]. In particular, a graduate student draws a stroke input for each image in the UCF test set, in a non-interactive fashion (if the user drew strokes interactively with any method, the results would be biased). Then, we obtain the shadow prediction results of: (a) the interactive shadow labeling method [18, 19]; (b) the closed-form matting method [36]; (c) our lazy annotation method (noisy labels). Finally, we apply the noisy label recovery method to obtain recovered labels. We also show the Stacked-DeconvNet results tested on the UCF test set to compare with.

The comparison results are shown in Table 2. Our noisy labels have the same BER as the results by Gong *et al.*[19]. Our recovered labels have a better BER, while our fully automatic Stacked-DeconvNet achieves the best performance, even without user input strokes.

**TABLE 2: Comparing to interactive shadow detection methods on the UCF test set [74].** Our noisy labels have the same BER compared to results by Gong *et al.*[19]. Our recovered labels have a better BER, and our fully automatic Stacked-DeconvNet achieves the best performance.

| Method | BER | Sha. | Non. |
|---|---|---|---|
| Interactive shadow labeling method [19] | 10.7 | 12.7 | 8.6 |
| Closed-form matting method [36] | 11.9 | 7.6 | 16.2 |
| Noisy labels | 10.7 | 14.2 | 7.1 |
| Recovered labels | 10.2 | 12.1 | 8.4 |
| Stacked-DeconvNet (fully automatic) | **9.4** | 10.2 | 8.6 |

### 7.3 Recovering added label noise on UCF dataset

To better evaluate our proposed label recovery method, we conducted experiments on the UCF dataset [74]. For the UCF training images, we have the clean shadow ground truth provided with the original dataset. Hence, we artificially add noise in the labeling by randomly flipping the provided labels. In this manner, we can evaluate the different facets of our label recovery method.

By randomly flipping some shadow region labels on the original UCF training set we create UCFNoisy. In this noisy version, $\sim$21% of the original ground-truth shadow pixels become dirty negatives (false non-shadow labels). See details in the No-Recovery row in Table 3. As we only flip some of the original labels for the shadow regions, the precision on shadow pixels is very high for No-Recovery (almost all of the pixels labeled as shadow are still shadow). Using this noisy version of the UCF dataset, we run our proposed label recovery method. We also run recovery with a linear kernel LSSVM (akin to our experiment in 7.1, but recovery is run on each cluster separately). Finally, we run the proposed $\mathcal{X}^2$ kernel LSSVM recovery method but on random equal size clusters. The number of clusters is 10, the same as the one obtained by our PGP-based clustering method.

**TABLE 3: Label recovery on noisy UCF data.** Comparison of our recovery method with $\mathcal{X}^2$ kernel versus linear kernel and random clustering. We measure precision and recall for shadow and non-shadow pixels by comparing the original ground-truth masks and the resulting masks for the different recovery methods. No Recovery denotes the results from comparing the noisy masks (UCFNoisy) with original UCF training set masks. Since UCFNoisy is created by flipping shadow labels, its recall for shadow pixels decreases significantly, while the precision remains high (almost all non-shadow pixels remain non-shadow).

| | | Sha. | | Non. | |
|---|---|---|---|---|---|
| Training Set | Method | Precision | Recall | Precision | Recall |
| UCFNoisy | No Recovery | 99.2 | 79.4 | 96.1 | 99.8 |
| UCFNoisy | Linear Kernel | 92.3 | 84.5 | 97.0 | 98.6 |
| UCFNoisy | Random Clusters | 94.5 | 87.0 | 97.5 | 99.0 |
| UCFNoisy | Proposed | **96.5** | **86.8** | **97.5** | **99.4** |

Table 3 shows the details for experiments on UCFNoisy. Our proposed method correctly recovers an extra 8% of shadow pixels, increasing the shadow pixel recall from 79.4% to 86.8%. That is, we appropriately flip 38% of the dirty negatives back to clean positives. Our method recovers these dirty negatives by slighty decreasing the precision on the shadow class by 2.7% (0.6% of the original non-shadow pixels are now flipped to be shadow, see Non-Shadow pixels Recall in Table 3). Recovery on random clusters, and recovery using a linear kernel perform worse than the proposed method. Overall, the proposed method correctly recovers most of the shadows with the highest precision.

### 7.4 Comparison to noise-tolerant methods

Our proposed method addresses label noise by focusing on recovering noisy labels, aiming for effective use of all training data. For completeness, we also compare to existing methods for classification with label noise. These methods in contrast, focus on being robust to noisy labels. These methods are designed to be unaffected by noisy labels rather than to effectively use noisy labels. These methods focus on asymptotic behavior with

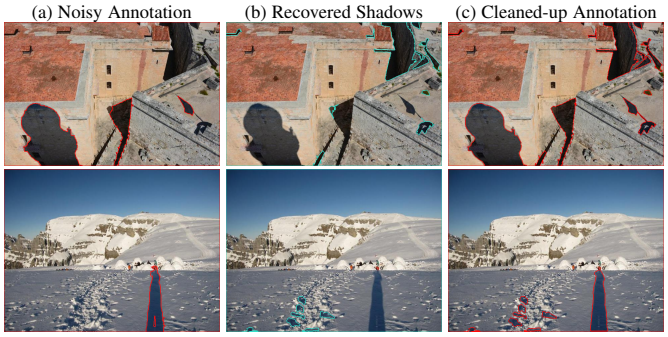



**Fig. 10: Recovery from noisy annotations.** Example of shadow region label recovery. a) Original shadow annotation depicted with red boundaries. b) Recovered shadows depicted with blue boundaries. c) Resulting cleaned-up shadow annotation: shadow boundaries depicted in red.

unlimited training data while our method focuses on the shadow detection problem where training data is very limited.

First, we implemented the noise-tolerant C-SVM method [40]. It achieves an average precision of 81.5 and 74.3, on the noisy UIUC and UCF datasets, respectively. These are significantly worse results than our method: 87.2 and 75.6, respectively, as seen in Table 1.

We then evaluate our method on the UCI dataset as in [40]. In these experiments, controlled levels of artificial noise are introduced in the labels. In Table 4 we report extensive comparison to several noise-tolerant methods. As we can see, our method is effective in leveraging noisy labels, even with significant levels of label noise ($\rho_+ = .4$ and $\rho_- = .4$). These results suggest that our method is more generally useful.

**TABLE 4: Classification accuracy of our method and several others on noisy UCI datasets.** $\rho_+, \rho_-$ are the portions of noisy positive and negative labels, respectively. Our method achieves highest or close to the highest accuracy for most datasets and noise levels. Entries within 1% from the best in each row are printed in bold.

| Data | $\rho_+$ | $\rho_-$ | $l_{\log}$[40] | C-SVM[40] | PAM[33] | NHERD[12] | RP[55] | Ours |
|---|---|---|---|---|---|---|---|---|
| Breast | .2 | .2 | 70.1 | 67.9 | 69.3 | 64.9 | 69.4 | **74.5** |
| | .3 | .1 | 70.1 | 67.8 | 67.8 | 65.7 | 66.3 | **74.0** |
| | .4 | .4 | 67.8 | 67.8 | 67.1 | 56.5 | 54.2 | **72.3** |
| Diabetes | .2 | .2 | **76.0** | 66.4 | 69.5 | 73.2 | 75.0 | **76.8** |
| | .3 | .1 | **75.5** | 66.4 | 65.9 | **74.7** | 67.7 | **75.1** |
| | .4 | .4 | 65.9 | 65.9 | 65.4 | **71.1** | 62.8 | 67.3 |
| Thyroid | .2 | .2 | 87.8 | 94.3 | **96.2** | 78.5 | 84.0 | 93.8 |
| | .3 | .1 | 80.3 | 92.5 | 86.9 | 87.8 | 83.1 | **95.6** |
| | .4 | .4 | 83.1 | 66.3 | 71.0 | 86.0 | 58.0 | **88.9** |
| German | .2 | .2 | 71.8 | 68.4 | 63.8 | 67.8 | 62.8 | **75.8** |
| | .3 | .1 | 71.4 | 68.4 | 67.8 | 67.8 | 67.4 | **77.7** |
| | .4 | .4 | 67.2 | 68.4 | 67.8 | 54.8 | 59.8 | **72.6** |
| Heart | .2 | .2 | **83.0** | 61.5 | 69.6 | **83.0** | 72.9 | 80.7 |
| | .3 | .1 | **84.4** | 57.0 | 62.2 | 81.5 | 79.3 | 79.7 |
| | .4 | .4 | 57.0 | 54.8 | 53.3 | 52.6 | 68.2 | **70.3** |
| Image | .2 | .2 | 82.5 | **92.0** | **92.9** | 77.8 | 65.3 | 91.3 |
| | .3 | .1 | 82.6 | **89.3** | **89.6** | 79.4 | 70.7 | 83.9 |
| | .4 | .4 | 63.5 | 63.5 | 73.2 | 69.6 | 64.7 | **81.9** |

### 7.5 Effect of noisy label recovery on CNN training

We call the training dataset of shadow images with noisy shadow masks labeled with user strokes SBU-Train-Noisy, and the training dataset with recovered masks as SBU-Train-Recover. For label recovery, PGP clusters SBU-Train-Noisy into 224 subsets of 10–60 images. To perform label recovery we allow up to 5% negative and up to 1% positive labels to be flipped ($\alpha = 0.99$, $\beta = 0.95$). We use our label recovery framework with a $\mathcal{X}^2$ kernel as the shadow region classifier. We choose the scaling parameter of the $\mathcal{X}^2$ kernel that minimizes the leave-one-out error on the noisy training set. We oversegment the training images into superpixels using Linear Spectral Clustering [73]. For each superpixel we compute intensity, color and texture features. We use 30-bin histograms for each of the channels of the CIELab color space. We represent texture with texton histograms. We run the full MR8 [58] filter bank on the input images and on the image density map [13]. Textons from density maps work well for shadow detection [13]. We cluster the filter responses, sampling 2,000 locations per image (balancing shadow and non-shadow pixels), to build two 128-word dictionaries. Our method is able to flip labels and correct some annotation mistakes. Fig. 10 shows examples of label recovery with new shadow boundaries in cyan.

Since we can not quantitatively evaluate our label recovery directly, we measure the influence of training with noisy versus recovered labels in terms of classification performance.

**TABLE 5: Label recovery influence on CNNs.** We show the BER of the FCN, the Patch-CNN, and the Stacked-FCN trained on SBU-Train-Noisy and SBU-Train-Recover, and tested on the UCF testing set and SBU-Test.

| Train Data | FCN | | Patch-CNN | | Stacked-FCN | |
|---|---|---|---|---|---|---|
| | UCF | SBU-Test | UCF | SBU-Test | UCF | SBU-Test |
| SBU-Train-Noisy | 20.0 | 17.7 | 14.1 | 12.6 | 14.0 | 12.1 |
| SBU-Train-Recover | **16.5** | **13.0** | **13.6** | **12.0** | **13.0** | **11.0** |

In Table 5, we compare the performance of the FCN, the Patch-CNN, and the stacked-CNN when trained on SBU-Train-Noisy and SBU-Train-Recover and tested on the UCF testing set and the proposed SBU-Test. The models trained with recovered labels outperform models trained with noisy labels. Using recovered labels reduces the error rate of the stacked-CNN by 7% and 9% respectively, when testing in UCF and SBU-Test. Similarly, label recovery reduces the error rate of the FCN by 17.5% and 26.5%.

## 8 Evaluation of Shadow Detection with Stacked CNN

In this section we evaluate our proposed shadow detection methods. For performance evaluation, we compare the predicted shadow masks with the high quality annotation/ground truth masks, measuring classification error rates at the pixel level. The main performance metric is the Balanced Error Rate (BER). We avoid an overall error metric because shadow pixels are considerably fewer than non-shadow pixels, hence classifying all pixels as non-shadow would yield a low overall error.

**CNN implementation details.** We use the FCN implementation by Long *et al.* [38]. We implement the Patch-CNN using Theano [3, 4]. For data augmentation during FCN and DeconvNet training, we downsample the training images by six different factors: 1.0, 0.9, 0.8, 0.7, 0.6, 0.5 and perform a left-right flip. For the Patch-CNN training, we store original images in memory and randomly extract patches on the fly, randomly rotate and flip them. The total training time of the stacked-CNN is approximately 10 hours on a single Titan X (maxwell) GPU.

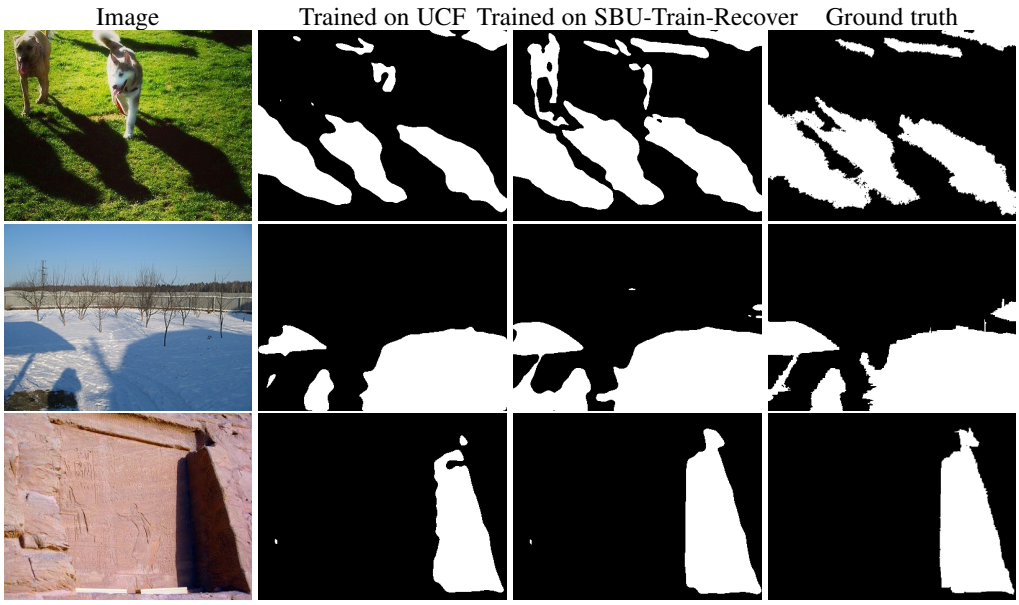



**Fig. 11: Comparison of Stacked-DeconvNet trained on UCF and SBU-Train-Recover.** A Stacked-DeconvNet trained on a larger dataset shows improved shadow segmentation compared to a Stacked-DeconvNet trained on the UCF training set. Since SBU-Train-Recover contains a larger variety of scenes, the classifier trained on it is more robust on a general test set.

**TABLE 6: Evaluation of shadow detection on UCF [74].** All methods are trained and tested on UCF training and test subsets. Stacked-DeconvNet achieves state-of-the-art level results.

| Method | BER | Sha. | Non. |
|---|---|---|---|
| Convnets+CRF [23] | 17.7 | 27.5 | 7.9 |
| LooKOP+MRF [59] | 13.2 | 20.0 | 6.4 |
| scGAN [41] | 10.9 | 10.4 | 11.4 |
| FCN [38] | 15.3 | 16.3 | 14.3 |
| Patch-CNN | 13.3 | 9.8 | 16.8 |
| Stacked-FCN | 11.6 | 10.4 | 12.8 |
| Stacked-DeconvNet | 9.4 | 10.2 | 8.6 |

### 8.1 Experiments on the UCF Dataset

We first evaluate our shadow detection method on the UCF dataset [74]. We train and test on the original UCF dataset (221 images), using the split given by Guo *et al*. [21]. Measuring performance in terms of BER, our proposed method (Stacked-DeconvNet) compares favorably to several state-of-the-art level methods†. Table 6 shows that our method achieves lower BER than ConvNets+CRF [23], and the kernel optimization method LooKOP+MRF [59], bringing a 34% and 12% error reduction, respectively. Fig. 12 shows some qualitative results of Stacked-FCN on the UCF test set.

### 8.2 Experiments with the SBU Dataset

To evaluate the generalization ability of our shadow detection method, we train our Stacked-CNNs on SBU-Train-Recover and test on the UCF testing set. As can be seen from Table 7, the Stacked-FCN and Stacked-DeconvNet trained on SBU-Train-Recover achieves lower error rate than LooKOP+MRF [59], which is trained on UCF. This suggests that our Stacked-CNNs trained on SBU-Train-Recover generalizes well to a totally different dataset. In Fig. 11, we show qualitative results comparing the performance of our Stacked-DeconvNet trained on UCF and SBU-Train-Recover datasets.

We also evaluate the performance of our proposed method on the newly collected testing set SBU-Test. Our Stacked-DeconvNet achieves 7.5% BER. Fig. 12 shows qualitative examples of shadow detection on SBU-Test. As can be seen, our proposed method is able to correctly identify shadows in a wide variety of scenes.

†. [53] cannot be directly compared because it used an extended version of the UCF dataset that is not publicly available.

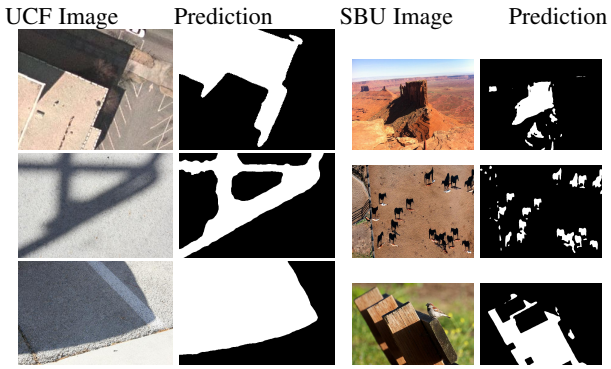



**Fig. 12: UCF and SBU qualitative results.** Examples of shadow detection by Stacked-DeconvNet on the UCF and SBU testing set.

### 8.3 Qualitative results on additional datasets

We show qualitative results on the Shadow Removal dataset [18, 19] and the Deshadow dataset [50] in Fig. 14, since shadow masks are not directly available for quantitative evaluation.

### 8.4 Alternative models for image level shadow prior

To further evaluate the efficacy of the proposed Stacked-CNN architecture, we compare each ILN architecture (FCN, CRF-RNN, SegNet, DeconvNet, ADNet), with its stacked counterpart (Stacked-FCN, Stacked-CRF-RNN, *etc*.). Results in Table 8 show that by stacking a Patch-CNN, we are able to improve

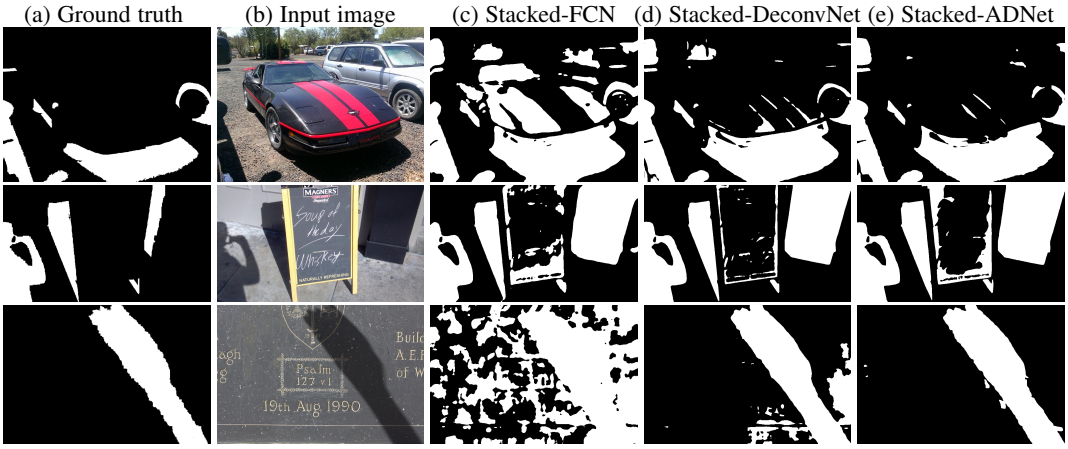



**Fig. 13: Shadow detection comparison between Stacked-FCN, Stacked-DeconvNet, and the Stacked-ADNet.** a) Ground truth shadow mask; b) Input image; c) Stacked-FCN prediction mask; d) Stacked-DeconvNet prediction mask; e) Stacked-ADNet prediction mask.

**TABLE 7: Results on the proposed SBU dataset and across UCF-SBU datasets.** We achieve state-of-the-art on the SBU-Train-Recover dataset. Training on SBU-Train-Recover dataset generalizes well on the UCF testing set, while the model trained on the UCF training set does not. ⋆: The DSC method [27] is tested on a different UCF test set which only has 76 images, whereas the other methods tested on a UCF test set with 110 images. †: the results of Deshadownet [50] are obtained by modifying the authors' code and retrain it for shadow detection.

| | | UCF-Test | | | SBU-Test | | |
|---|---|---|---|---|---|---|---|
| Training Set | Method | BER | Sha. | Non. | BER | Sha. | Non. |
| SBU-Train-Recover | scGAN [41] | - | - | - | 11.5 | 7.7 | 15.3 |
| | DSC [27]⋆ | 8.1 | - | - | 5.6 | - | - |
| | ADNet [35] | - | - | - | 5.4 | 5.3 | 5.5 |
| | Deshadownet [50]† | - | - | - | 10.2 | 8.3 | 12.0 |
| UCF Train | Stacked-FCN | 11.6 | 10.4 | 12.8 | 13.9 | 13.1 | 14.7 |
| SBU-Train-Noisy | Stacked-FCN | 14.0 | - | - | 12.1 | - | - |
| SBU-Train-Recover | Stacked-FCN | 13.0 | 9.0 | 17.1 | 11.0 | 9.6 | 12.5 |
| | Stacked-DeconvNet | 10.3 | 8.7 | 11.9 | 7.5 | 4.9 | 10.1 |
| | Stacked-ADNet | **9.2** | 8.4 | 10.0 | **4.7** | 3.7 | 5.8 |

**TABLE 8: Image-level shadow predictor network (ILN) comparison.** Balanced Error Rate (BER) of shadow prior models on the UCF and SBU datasets.

| Method | UCF-Test | SBU-Test |
|---|---|---|
| FCN | 15.3 | 13.0 |
| Stacked-FCN | 11.6 | 11.0 |
| FCN+CRF-RNN | 12.5 | 13.7 |
| Stacked-FCN+CRF-RNN | 10.6 | 10.4 |
| SegNet | 9.9 | 7.9 |
| Stacked-SegNet | 9.8 | 7.1 |
| DeconvNet+CRF | 10.5 | 8.3 |
| Stacked-DeconvNet | **9.4** | 7.5 |
| ADNet | - | 5.4 |
| Stacked-ADNet | - | **4.7** |

any semantic segmentation network consistently. Additionally, we achieve state-of-the-art performance on the SBU dataset (named as SBU-Train-Recover in Table 8). Fig. 13, shows a qualitative comparison of shadow prediction results between Stacked-FCN, Stacked-DeconvNet, and Stacked-ADNet. Detection improves on

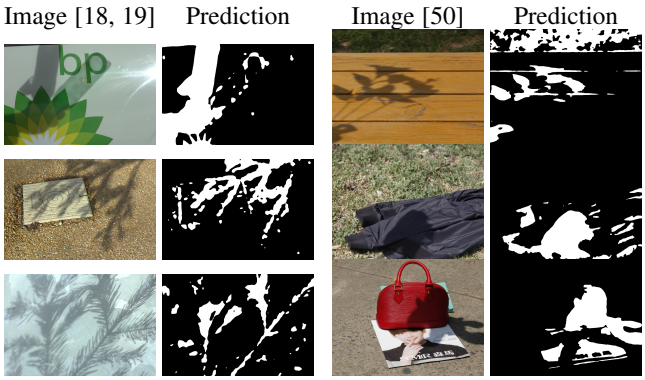



**Fig. 14: Qualitative results on the Shadow Removal dataset [18, 19] (left) and the Deshadow dataset [50] (right),** given by Stacked-DeconvNet.

materials with challenging reflectance properties such as dark materials.

Although DeconvNet [42], SegNet [2], and the U-net based scGAN [41] maintain spatial information via unpooling layers or skip connections, due to the capacity of the networks, segmentation results do not stick to shadow boundaries perfectly. As a postprocessing step, CRFs were used [10, 23] to make the prediction results more consistent with local texture and color. The proposed patch-CNN can be viewed as a very high order CRF-like method, for the same purpose. By comparing the stacked-CNN against the DeconvNet baseline which has a CRF postprocessing step, and against the CRF-RNN baseline which has a built-in CRF in this Section, we see that adding the proposed patch-CNN instead of CRFs yield significantly better performance.

## 9 Conclusions

We have introduced lazy annotation, a framework for efficient collection of annotated shadow datasets. We have shown how to leverage the noisy labels through a label recovery process. This process is efficient as it is based on minimizing the leave-one-out error of Least Squares SVM. Our experiments show that when training with recovered labels, the performance penalty is small.

We extended our method to perform large-scale label recovery of noisily annotated shadow regions. This allowed us to create a new shadow dataset that is 20 times bigger than existing datasets. This dataset is well suited for deep-learning, and we proposed

a novel deep learning framework to take advantage of the new dataset. Our deep learning architecture operates at the local patch level, but it can incorporate the global semantics through an image level shadow prior. This leads to a shadow classifier that performs well across different datasets. We have shown that our proposed framework can successfully integrate different semantic segmentation models adapting them as image-level shadow prior generators.

The proposed dataset is already being used by recent shadow detection methods [26, 27, 35, 39, 41, 45, 64, 65, 67, 71, 75], and we expect it to become the benchmark for shadow detection. We also plan to adapt our method to combine datasets collected under different annotation methodologies. Such datasets would contribute to the progress of shadow detection and scene understanding. Furthermore, we will explore generalizing label recovery to other domains.

### Acknowledgement

We thank Hieu Le for providing results of the A+D Net [35]. This work is partially supported by NSF IIS-1161876 and CNS-1718014, FRA DTFR5315C00011, the Stony Brook SensonCAT, the Subsample project from DIGITEO Institute, France, a gift from Adobe, the Partner University Fund, and the SUNY2020 Infrastructure Transportation Security Center. The authors would like to thank Amazon for providing EC2 credits and NVIDIA for donating GPUs.

## REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE PAMI*, 34(11):2274–2281, 2012.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE PAMI*, 39(12):2481–2495, 2017.

[3] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. NIPS Workshop, 2012.

[4] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *SciPy*, 2010.

[5] B. Biggio, B. Nelson, and P. Laskov. Support vector machines under adversarial label noise. In *ACML*, 2011.

[6] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proc. ICCV*, 2001.

[7] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proc. ECCV*, 2008.

[8] J. Canny. A computational approach to edge detection. *IEEE PAMI*, 1986.

[9] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. *arXiv*, 2015.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[11] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proc. CVPR*, 2012.

[12] K. Crammer and D. D. Lee. Learning via gaussian herding. In *NIPS*, 2010.

[13] A. Ecins, C. Fermller, and Y. Aloimonos. Shadow-free segmentation in still images using local density measure. In *Intl. Conf. Image Proc.*, 2014.

[14] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2012 (voc2012) results (2012). In *URL http://www. pascal-network. org/challenges/VOC/voc2011/workshop/index. html*, 2011.

[15] G. Finlayson, M. Drew, and C. Lu. Entropy minimization for shadow removal. *IJCV*, 85:35–57, 2009.

[16] G. Finlayson, S. Hordley, C. Lu, and M. Drew. On the removal of shadows from images. *IEEE PAMI*, 28(1):59–68, 2006.

[17] B. Frenay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, May 2014.

[18] H. Gong and D. Cosker. Interactive shadow removal and ground truth for variable scene categories. In *BMVC*, 2014.

[19] H. Gong and D. Cosker. Interactive removal and ground truth for difficult shadow scenes. *JOSA A*, 33(9), 2016.

[20] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Proc. CVPR*, 2010.

[21] R. Guo, Q. Dai, and D. Hoiem. Single-image shadow detection and removal using paired regions. In *Proc. CVPR*, 2011.

[22] R. Guo, Q. Dai, and D. Hoiem. Paired regions for shadow detection and removal. *IEEE PAMI*, 35(12):2956–2967, 2012.

[23] S. Hameed Khan, M. Bennamoun, F. Sohel, and R. Togneri. Automatic feature learning for robust shadow detection. In *Proc. CVPR*, 2014.

[24] M. Hoai. Regularized max pooling for image categorization. In *Proc. BMVC*, 2014.

[25] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *Proc. ACCV*, 2014.

[26] S. Hosseinzadeh, M. Shakeri, and H. Zhang. Fast shadow detection from a single image using a patched convolutional neural network. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3124–3129. IEEE, 2018.

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

[28] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE PAMI*, 2013.

[29] X. Jiang, A. Schofield, and J. Wyatt. Shadow detection based on colour segmentation and estimated illumination. In *Proc. BMVC*, 2011.

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

[31] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. CVPR*, 2014.

[32] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. *ACM Trans. Graph.*, 2011.

[33] R. Khardon and G. Wachman. Noise tolerant variants of the perceptron algorithm. *J. Machine Learning Research*, 8:227–248, May 2007.

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

[35] 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 *Proc. ECCV*, 2018.

[36] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2008.

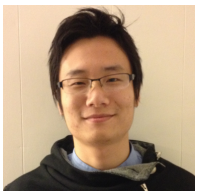
[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, 2014.

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

[39] S. Mohajerani and P. Saeedi. Shadow detection in single rgb images using a context preserver convolutional neural network trained by multiple adversarial examples. *IEEE Transactions on Image Processing*, 2019.

[40] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari.

Learning with noisy labels. In *NIPS*, 2013.
[41] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, and D. Samaras. Shadow detection with conditional generative adversarial networks. In *ICCV*, 2017.
[42] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proc. ICCV*, 2015.
[43] I. Okabe, T. Sato and Y. Sato. Attached shadow coding: estimating surface normals from shadows under unknown reflectance and lighting conditions. In *Proc. ECCV*, 2009.
[44] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation fo the spatial envelope. *IJCV*, 2001.
[45] T. Pan, B. Wang, G. Ding, and J.-H. Yong. Shadow detection using robust texture learning. In *BMVC*, page 114, 2018.
[46] A. Panagopoulos, D. Samaras, and N. Paragios. Robust shadow and illumination estimation using a mixture model. In *Proc. CVPR*, 2009.
[47] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios. Simultaneous cast shadows, illumination and geometry inference using hypergraphs. *IEEE PAMI*, 2013.
[48] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. *arxiv*, 2015.
[49] E. Park, X. Han, T. L. Berg, and A. C. Berg. Combining multiple sources of knowledge in deep cnns for action recognition. In *WACV*, 2016.
[50] L. Qu, J. Tian, S. He, Y. Tang, and R. W. Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *CVPR*, 2017.
[51] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proc. ICML*, 1998.
[52] A. J. Sharkey. *Combining artificial neural nets: ensemble and modular multi-net systems*. Springer Science & Business Media, 2012.
[53] L. Shen, T. W. Chua, and K. Leman. Shadow optimization from structured deep edge detection. In *Proc. CVPR*, 2015.
[54] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.
[55] G. Stempfel and L. Ralaivola. Learning kernel perceptrons on noisy data using random projections. *Algorithmic Learning Theory*, pages 328–342, 2007.
[56] G. Stempfel and L. Ralaivola. Learning svms from sloppily labeled data. In *International Conference on Artificial Neural Networks*. 2009.
[57] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.
[58] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proc. ECCV*, 2002.
[59] T. F. Y. Vicente, M. Hoai, and D. Samaras. Leave-one-out kernel optimization for shadow detection. In *Proc. ICCV*, 2015.
[60] T. F. Y. Vicente, M. Hoai, and D. Samaras. Noisy label recovery for shadow detection in unfamiliar domains. In *Proc. CVPR*, 2016.
[61] T. F. Y. Vicente, M. Hoai, and D. Samaras. Leave-one-out kernel optimization for shadow detection and removal. *IEEE PAMI*, 40(3):682–695, 2018.
[62] 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 *Proc. ECCV*, 2016.
[63] T. F. Y. Vicente, C.-P. Yu, and D. Samaras. Single image shadow detection using multiple cues in a supermodular MRF. In *Proc. BMVC*, 2013.
[64] J. Wang, X. Li, and J. Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *CVPR*, 2018.
[65] 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.
[66] Z. Wei and M. Hoai. Region ranking svms for image classification. In *Proc. CVPR*, 2016.
[67] Z. Wu, L. Su, and Q. Huang. Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2019.
[68] C.-P. Yu, W.-Y. Hua, D. Samaras, and G. Zelinsky. Modeling clutter perception using parametric proto-object partitioning. In *NIPS*, 2013.
[69] C.-P. Yu, H. Le, G. Zelinsky, and D. Samaras. Efficient video segmentation using parametric graph partitioning. In *ICCV*, 2015.
[70] L. Zhang, Q. Zhang, and C. Xiao. Shadow remover: Image shadow removal based on illumination recovering optimization. *IEEE Transactions on Image Processing*, 24(11):4623–4636, 2015.
[71] Q. Zheng, X. Qiao, Y. Cao, and R. W. Lau. Distraction-aware shadow detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5167–5176, 2019.
[72] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *Proc. ICCV*, 2015.
[73] L. Zhengqin and C. Jiansheng. Superpixel segmentation using linear spectral clustering. In *Proc. CVPR*, 2015.
[74] J. Zhu, K. Samuel, S. Masood, and M. Tappen. Learning to recognize shadows in monochromatic natural images. In *Proc. CVPR*, 2010.
[75] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 121–136, 2018.
[76] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004.



**Le Hou** is a PhD. candidate working with Prof. Dimitris Samaras in the Computer Vision Lab at Stony Brook University. Previously, he worked as a senior software engineer at Baidu INC, after he graduated as a Bachelor of Computer Science and Technology from Huazhong University of Science and Technology.

**Tomás F. Yago Vicente** is a PhD. at Computer Sciences and a software engineer at A9.com. He obtained his PhD. degree at Stony Brook University, before which he was affiliated with the 3D Group for Interactive Visualization at the University of Rhode Island. He received the Computer Engineer diploma from University of Zaragoza, Spain in 2008.

**Minh Hoai** is an Assistant Professor of Computer Science at Stony Brook University and a Principal Research Scientist at VinAI Research. He received a Bachelor of Software Engineering from the University of New South Wales in 2005 and a Ph.D. in Robotics from Carnegie Mellon University in 2012. His research interests are in computer vision and machine learning, especially human action and activity recognition and prediction.

**Dimitris Samaras** received a diploma degree in computer science and engineering from the University of Patras in 1992, the MSc degree from Northeastern University in 1994, and the PhD degree from the University of Pennsylvania in 2001. He is an SUNY Empire Innovation professor of Computer Science at Stony Brook University. He is the Program Chair of CVPR 2022. His research interests lie in 3D shape and motion estimation for human behavior analysis, illumination modeling and estimation for recognition and graphics, and biomedical image analysis.