

# Noisy Label Recovery for Shadow Detection in Unfamiliar Domains

Tomás F. Yago Vicente, Minh Hoai, Dimitris Samaras  
Stony Brook University, Stony Brook, NY 11794, USA

{tyagovicente, minhhoai, samaras}@cs.stonybrook.edu

## 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. This is due to the intense manual labor in 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 the label recovery problem 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 suggest a feasible approach to address the task of detecting shadows in an unfamiliar domain: collecting and lazily annotating some images from the new domain for training. As will be demonstrated, this approach outperforms methods that rely on precisely annotated but less relevant datasets. Initial results suggest more general applicability.*

## 1. Introduction

The problem of single image shadow detection has been widely studied. Early work such as the illumination invariant approaches [5, 6] are based on physical modeling of the illumination and shadowing phenomena [20, 21]. These physics-based methods only work well with high quality images. In contrast, statistical learning approaches (e.g., [9, 15, 23, 29, 31]) have shown significant success in detecting shadows in consumer-grade photos and web quality images. The performance of these methods, however, depend on the quality and quantity of training images. Guo et al. [9], Zhu et al. [31] were the firsts to collect sizable datasets of images with annotated shadows, which are referred to as

the UCF and UIUC datasets respectively. These two publicly available datasets have been used to develop several shadow detection methods [13, 14, 17, 28, 29], and staggering progress has been made in the past few years. However, these two datasets are small, and the methods trained on them do not generalize well to new domains (e.g., see [10] for poor cross-dataset performance analysis). Unfortunately, there is no larger and publicly available shadow datasets; this is perhaps due to the huge effort required to properly annotate shadows in images.

In this paper we propose “lazy annotation”, a method that allows a human annotator to quickly label images to create shadow datasets. The annotator needs to focus only on the most relevant shadows in an image and draw several strokes on the perceived important shadow and non-shadow areas of the image. We process the input strokes to segment shadow areas based on image features using the geodesic convexity image segmentation [8]. Figure 1 shows an example of this process, from the annotator’s strokes to the generated binary shadow mask.

With lazy annotation, it is possible to quickly collect shadow annotation. The annotation, however, is imperfect. Due to the nature of the task, some shadow areas may be ignored, or imperfectly segmented, as shown in Figure 1. The resulting annotated data presents noticeable class label noise; we refer to shadow and non-shadow regions as positive and negative classes respectively. Label noise is asymmetric. The negative class contains “dirty negatives”, corresponding to missed shadows, or poorly segmented regions that contain both shadow and non-shadow pixels. The positive class is significantly cleaner and more reliable, because the annotator is asked to label some shadows, so the shadow regions obtained are generally well segmented.

The presence of label noise in training data has huge impact on the performance of classifiers trained on the data (e.g., see [7, 32]). To address the problem of noisy labels, we propose to jointly learn a shadow region classifier and recover the labels in the training set. Our objective is to reduce the level of label noise in the training set so as to minimize the generalization error of the learned classifier. Our framework is based on Least-Squares Support Vector Ma-

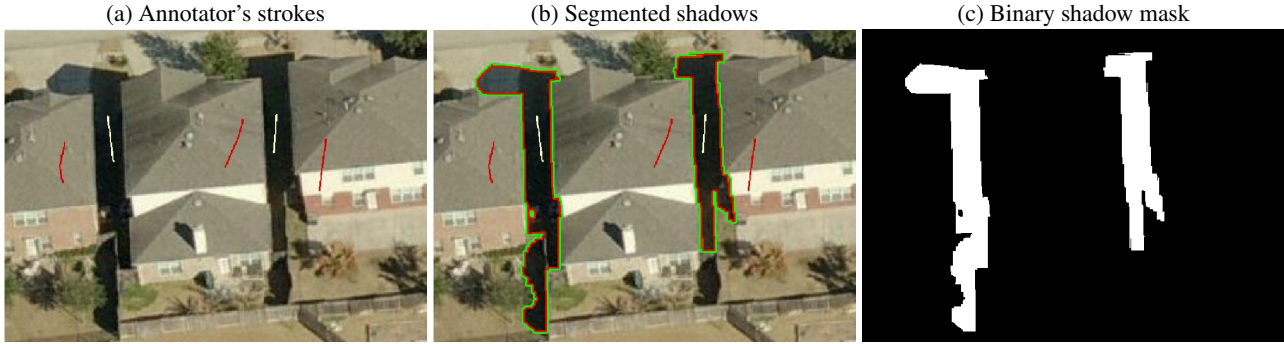


Figure 1. **Lazy annotation pipeline for efficient labeling of shadow images.** a) An annotator is asked to draw some strokes on some (not all) shadow areas (white strokes) and non-shadow areas (red strokes). b) Automatically segmented shadow regions. c) Obtained shadow mask, mostly good with a few exceptions 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.

chines (LSSVM). LSSVM has a closed-form solution, and the leave-one-out error of LSSVM is a linear function of the training labels. To jointly learn the classifier and recover the labels, we 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 can constrain the fraction of originally labeled positive and negative instances that are flipped.

To validate our approach, we “lazily annotated” the UIUC and UCF training sets. Experimental results show that a classifier trained with recovered labels achieved 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 by 10% and 3%, in the UIUC and UCF datasets respectively. We show experimentally that label recovery is robust up to significant levels of label noise in the training set.

We also present here a new test set with carefully annotated shadow labels. We collected images from a wide variety of scene types, some of which, such as snow or beach, significantly differ from UCF and UIUC contents. As expected, cross dataset performance of models trained in UCF or UIUC is reduced in comparison to their respective test sets. We also collected a new training set that had some (but not all) similar scenes to the test set and obtained lazy labels for it. Experimental results show that the classifier trained on the lazily annotated set performs better than the models trained on UCF and UIUC datasets with more accurate labels. Using the recovered lazy labels further boosts the classifier performance by an extra 3.3%. We show some initial promising results that suggest that our method generalizes to other domains that suffer from label noise.

In summary, we make the following contributions:

- 1) Propose a new method for fast collection of reasonably good shadow annotations.
- 2) Propose a new framework that jointly recovers the true labels of imperfect annotations and learns a classifier.

- 3) Demonstrate that efficiently labeling a new training dataset is better than using models trained on accurately labeled datasets with different scene characteristics.

## 2. Previous Work

### 2.1. Annotated shadow datasets

Annotated shadow datasets fostered work on shadow detection. However, there are only a few shadow datasets due to the cumbersome nature of the annotation process. Shadow annotation is painstaking: the human annotator has to first 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.

Guo et al. [9] 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 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.

Existing publicly available shadow datasets are small, and the methods trained on them do not perform well on other datasets. Guo et al. [9, 10] report the cross-dataset performance of their model for UIUC and UCF datasets. The results are alarming but not surprising: a model trained on UCF training set performs well on the UCF test set, but not on the UIUC test set (90.2% versus 81.5% accuracy), and a model for UIUC dataset has much less accuracy when tested on the UCF test set (10.7% reduction in accuracy).

## 2.2. Noisy label recovery

The presence of label noise lowers the performance of classification tasks [7, 32]. Recent methods that address the problem of label noise in the data [2, 19, 24, 25] aim to be robust to noisy labels, and 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[24], class-dependent with fixed probability [2], limited noise ratios[19].

## 2.3. Review of Least Squares SVM

Our framework for recovering noisy annotation and training a classifier is based on Least-Squares Support Vector Machines (LSSVM) [26], which is also known as ridge regression [22]. LSSVM has a closed-form solution, which is a computational advantage over SVM. Furthermore, once the LSSVM solution has been computed, the solution for a reduced training set, obtained by removing any training data point, can be found efficiently. This enables reusing training data for further calibration, e.g., [11, 12, 29, 30]. This also enables using the training data for correcting the noisy labels, as proposed in Section 4. This section reviews LSSVM and the leave-one-out formula.

Given a training set of  $n$  data points  $\{\mathbf{x}_i\}_{i=1}^n$  and associated labels  $\{y_i | y_i \in \{0, 1\}\}_{i=1}^n$ , LSSVM optimizes the following:

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

Here  $s_i$  is the instance weight, allowing the assignment of different weights to different training instances. Let  $\bar{\mathbf{X}} = [\mathbf{X}; \mathbf{1}_n^T]$ ,  $\bar{\mathbf{w}} = [\mathbf{w}, b]$ ,  $\bar{\mathbf{I}} = [\mathbf{I}_{n \times n}, \mathbf{0}_n; \mathbf{0}_n^T, 0]$ . Eq. (1) is equivalent to:

$$\underset{\bar{\mathbf{w}}}{\text{minimize}} \quad \lambda \bar{\mathbf{w}}^T \bar{\mathbf{I}} \bar{\mathbf{w}} + \sum_{i=1}^n s_i (\bar{\mathbf{w}}^T \bar{\mathbf{x}}_i - y_i)^2. \quad (2)$$

This is an unconstrained convex quadratic problem, and the optimal solution is attained where the gradient is zero. The gradient of Eq. (2) with respect to  $\bar{\mathbf{w}}$  is:

$$2\lambda \bar{\mathbf{I}} \bar{\mathbf{w}} + \sum_{i=1}^n 2s_i (\bar{\mathbf{w}}^T \bar{\mathbf{x}}_i - y_i) \bar{\mathbf{x}}_i \quad (3)$$

$$= 2(\bar{\mathbf{X}} \text{diag}(\mathbf{s}) \bar{\mathbf{X}}^T + \lambda \bar{\mathbf{I}}) \bar{\mathbf{w}} - 2\bar{\mathbf{X}} \text{diag}(\mathbf{s}) \mathbf{y} \quad (4)$$

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

Let  $\mathbf{C} = (\bar{\mathbf{X}} \text{diag}(\mathbf{s}) \bar{\mathbf{X}}^T + \lambda \bar{\mathbf{I}})$  and  $\mathbf{d} = \bar{\mathbf{X}} \text{diag}(\mathbf{s}) \mathbf{y}$ , the optimal solution can be found by setting the gradient to zero, leading to the close-form solution:  $\bar{\mathbf{w}} = \mathbf{C}^{-1} \mathbf{d}$ .

Now, suppose we remove the training instance  $\mathbf{x}_i$ , let  $\mathbf{C}_{(i)}$ ,  $\mathbf{d}_{(i)}$ ,  $\mathbf{w}_{(i)}$  be the corresponding values for removing  $\mathbf{x}_i$ . We have:  $\bar{\mathbf{w}}_{(i)} = \mathbf{C}_{(i)}^{-1} \mathbf{d}_{(i)}$ , where  $\mathbf{C}_{(i)} = \mathbf{C} - s_i \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T$  and  $\mathbf{d}_{(i)} = \mathbf{d} - y_i s_i \bar{\mathbf{x}}_i$ . Using the Sherman–Morrison formula, we have:

$$\mathbf{C}_{(i)}^{-1} = (\mathbf{C} - s_i \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T)^{-1} = \mathbf{C}^{-1} + \frac{\mathbf{C}^{-1} s_i \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T \mathbf{C}^{-1}}{1 - s_i \bar{\mathbf{x}}_i^T \mathbf{C}^{-1} \bar{\mathbf{x}}_i}.$$

Substituting the above equation to  $\bar{\mathbf{w}}_{(i)} = \mathbf{C}_{(i)}^{-1} \mathbf{d}_{(i)}$  and developing the derivation, we get:

$$\bar{\mathbf{w}}_{(i)} = \bar{\mathbf{w}} + s_i (\mathbf{C}^{-1} \bar{\mathbf{x}}_i) \frac{-y_i + \bar{\mathbf{x}}_i^T \bar{\mathbf{w}}}{1 - s_i \bar{\mathbf{x}}_i^T \mathbf{C}^{-1} \bar{\mathbf{x}}_i}. \quad (5)$$

Therefore, the LOO error is:

$$\begin{aligned} \bar{\mathbf{w}}_{(i)}^T \bar{\mathbf{x}}_i - y_i &= \bar{\mathbf{w}}^T \bar{\mathbf{x}}_i - y_i + s_i \bar{\mathbf{x}}_i^T \mathbf{C}^{-1} \bar{\mathbf{x}}_i \frac{-y_i + \bar{\mathbf{x}}_i^T \bar{\mathbf{w}}}{1 - s_i \bar{\mathbf{x}}_i^T \mathbf{C}^{-1} \bar{\mathbf{x}}_i} \\ &= \frac{\bar{\mathbf{w}}^T \bar{\mathbf{x}}_i - y_i}{1 - s_i \bar{\mathbf{x}}_i^T \mathbf{C}^{-1} \bar{\mathbf{x}}_i} \end{aligned} \quad (6)$$

In summary, let  $\mathbf{M} = \mathbf{C}^{-1} \bar{\mathbf{X}}$  and  $\mathbf{H} = \mathbf{M}^T \bar{\mathbf{X}}$ , then:

$$\text{The weight vector : } \bar{\mathbf{w}} = \mathbf{M} \text{diag}(\mathbf{s}) \mathbf{y} \quad (7)$$

$$\text{LOO weight vector: } \bar{\mathbf{w}}_{(i)} = \bar{\mathbf{w}} + \frac{(\bar{\mathbf{w}}^T \bar{\mathbf{x}}_i - y_i) s_i}{1 - s_i h_{ii}} \mathbf{m}_i \quad (8)$$

$$\text{LOO error} = \bar{\mathbf{w}}_{(i)}^T \bar{\mathbf{x}}_i - y_i = \frac{\bar{\mathbf{w}}^T \bar{\mathbf{x}}_i - y_i}{1 - s_i h_{ii}} \quad (9)$$

## 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. This task is a painstaking process with a heavy burden on the annotator. We simplify the annotation task by redefining its goal. Rather than aiming to segment 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 area. 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. The annotator strokes are processed with the segmentation method of Gulshan et al. [8] to generate a binary mask.

### 3.1. Lazy annotation pipeline

We illustrate our lazy annotation pipeline with an example image in Figure 2. First, the annotator is instructed to

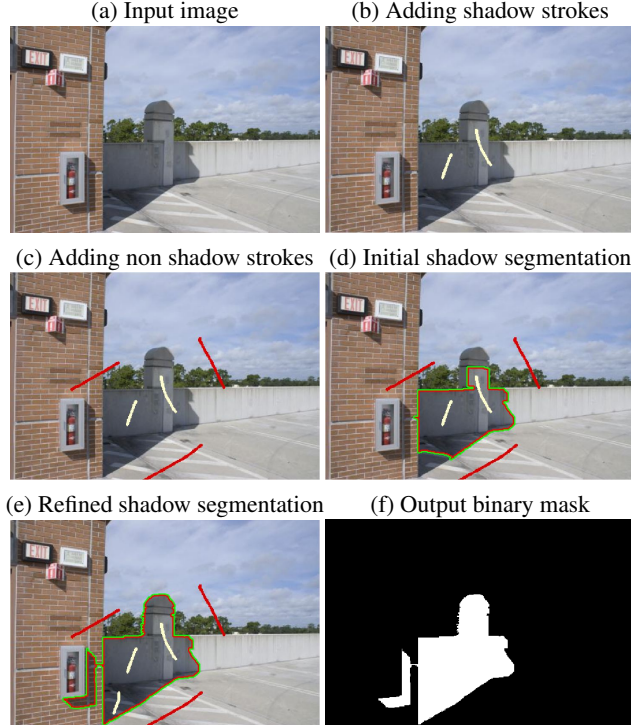


Figure 2. **Lazy annotation pipeline.** a) Input image. b) Annotator’s shadow strokes in white. c) Annotator’s non shadow strokes in red. d) Initial shadow segmentation in green (outer side) and red (inner side). e) Refined shadow segmentation with a final shadow stroke in the lower center of the image.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

draw a few strokes (2-3) on areas of the image she considers relevant shadows, see Figure 2.b. Then, the annotator draws a few strokes (2-3) on non shadow areas surrounding the shadow, see Figure 2.c. After that, a shadow segmentation based on the strokes is presented to the annotator, see Figure 2.d. Then, the annotator is able to add a few additional strokes to refine the shadow segmentation interactively. In Figure 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 Figure 2.f.

### 3.2. Annotation tool

We interactively segment the images using the method of Gulshan et al. [8]. The method combines geodesic star convexity shape constraints with the Boykov-Jolly [3] 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 [8] to render a more streamlined user interface tailored for 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 is typically able to label an average of 3 images a minute.

### 3.3. Postprocessing

In this work, 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 presented by Vicente et al. [28] for shadow detection. First, we oversegment the image into SLIC [1] superpixels (see Figure 3.a). Then, we apply Mean-shift clustering in Lab space and merge connected superpixels in the same cluster into a larger region, see Figure 3.b.

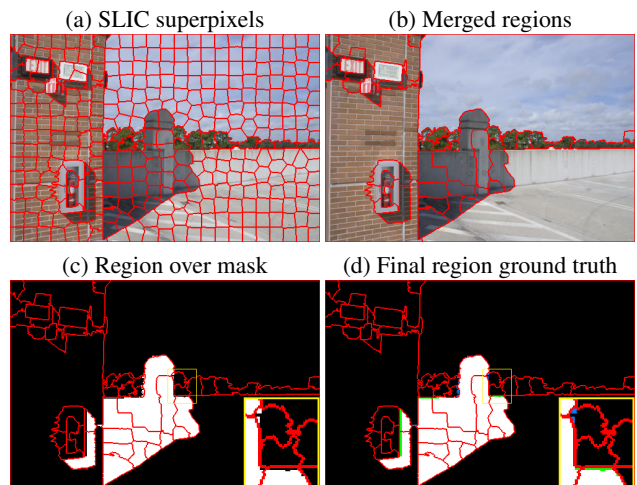


Figure 3. **From lazy shadow mask to region labels.** a) Initial SLIC superpixels. b) Regions obtained by merging superpixels. c) Lazy mask overlaid on regions. d) Final region ground-truth.

We overlay the binary mask on the segmented regions (Figure 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 Figure 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

We pose noisy label recovery as an optimization problem where the labels of some training examples can be flipped to minimize the sum of squared leave-one-out errors. Our formulation exploits the fact that the leave-one-out error of LSSVM (Sec. 2.3) is a linear function of the labels.

### 4.1. Formulation

Reconsider the formula for the leave-one-out error given in Eq. (9), substituting the formula for  $\bar{w}$  given in Eq. (7),

the leave-one-out error is:

$$\bar{\mathbf{w}}_{(i)}^T \bar{\mathbf{x}}_i - y_i = \frac{\bar{\mathbf{x}}_i^T \mathbf{M} \text{diag}(\mathbf{s}) \mathbf{y} - y_i}{1 - s_i h_{ii}} \quad (10)$$

Let  $\mathbf{P} = \text{diag}(\mathbf{s})\mathbf{H}$  and recall  $\mathbf{H} = \mathbf{M}^T \bar{\mathbf{X}}$  (Section 2.3). The leave-one-out error can be shown to be:

$$\bar{\mathbf{w}}_{(i)}^T \bar{\mathbf{x}}_i - y_i = \frac{\mathbf{p}_i^T \mathbf{y} - y_i}{1 - p_{ii}} \quad (11)$$

Let  $\mathbf{e}_i$  be the  $i^{\text{th}}$  column of the identity matrix of size  $n$ , and let  $\mathbf{a}_i = \frac{\mathbf{p}_i - \mathbf{e}_i}{1 - p_{ii}}$ , the leave-one-out error becomes: .

$$\bar{\mathbf{w}}_{(i)}^T \bar{\mathbf{x}}_i - y_i = \frac{\mathbf{p}_i^T \mathbf{y} - y_i}{1 - p_{ii}} = \mathbf{a}_i^T \mathbf{y}. \quad (12)$$

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 indexes of (noisy) positive and negative training instances respectively, i.e.,  $\mathcal{P} = \{i | y_i = 1\}$  and  $\mathcal{N} = \{i | y_i = 0\}$ . Our noisy label recovery minimizes the sum of squared leave-one-out errors:

$$\underset{\mathbf{y}}{\text{minimize}} \sum_{i=1}^n (\mathbf{a}_i^T \mathbf{y})^2 \quad (13)$$

$$\text{s.t.} \sum_{i \in \mathcal{P}} y_i \geq \alpha |\mathcal{P}|, \quad (14)$$

$$\sum_{i \in \mathcal{N}} y_i \leq (1 - \beta) |\mathcal{N}|, \quad (15)$$

$$y_i \in \{0, 1\}. \quad (16)$$

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 ( $0 \leq \alpha, \beta \leq 1$ ). Constraint (14) requires the proportion of original positive training instances that remain positive must be greater than or equal  $\alpha$ . If  $\alpha = 1$  none of the positive instances can become negative. Similarly, Constraint (15) limits the proportion of flipped negative data points to be at most  $1 - \beta$ .

## 4.2. Optimization

The optimization problem in Eq. (13) is a quadratic program with linear constraints and binary variables, that can be optimized in two steps. First, we relax the binary constraints to find a relaxed solution for  $\mathbf{y}$  where the entries are between 0 and 1 instead of being either 0 or 1. Second, starting from the relaxed solution, we perform block coordinate descent to find an optimal binary solution. In fact, even though the first step is a quadratic program with linear constraints, we also optimize it by block coordinate descent, as this is more efficient than solving the entire problem at once.

For the optimization problem in either Step 1 or Step 2 (with and without binary constraints), block coordinate descent works as follows. We run the optimization with multiple epochs, each epoch is a complete pass through all training data. For each epoch, we randomly divide the training data into multiple batches of a desired batch size. Considering each batch in turn, we optimize a sub problem that is obtained by fixing all the variables not in the batch. Once we have visited all batches, we recalculate the objective value and compare it with the objective obtained from the last epoch. If the objective value is not significantly reduced, we terminate the optimization procedure.

Block coordinate descent is guaranteed to converge, because each step of the optimization does not increase the objective value. The optimization problem in Step 1 is convex, and so block coordinate descent will converge to the global solution. The optimization problem in Step 2 is not convex, but the initial starting position is relatively good so it will likely converge to a reasonable solution.

## 5. Experiments and Results

We conducted experiments on the UCF [31] and the UIUC [9] datasets. We also compiled a new dataset from publicly available images, which will we refer to as the SBU dataset. About half of the images come from the Microsoft COCO [18] dataset, and the other half from the Web. The SBU dataset consists of 210 images depicting a wide variety of scenes such as: urban, roads, beach, snowy, horses, planes, people playing tennis, parks. We created a dataset for testing cross-dataset performance for multiple scene types. We split the dataset evenly into training and testing sets. The training set is labeled using the proposed lazy annotation approach, whereas the test set is carefully annotated. The dataset is available at <http://www3.cs.stonybrook.edu/~cvl/dataset.html>.

On all of our experiments, the region classifier is a Least Squares SVM with a linear kernel. We use texture, color and intensity features. For each region, we compute a tex-ton histogram (using full MR8 filters [27]), and a histogram on each component of the CIELAB color space. We measure the performance of the classification task in terms of Average Precision (AP). Since each data point corresponds to a region, we weight each region by its area in pixels to approximate pixel AP.

### 5.1. Shadow detection on a new domain

To study shadow detection on a new domain, we train a shadow region classifier on the UCF and UIUC training sets. Notice that these sets contain carefully annotated labels. We then test the classifier on the newly collected SBU test set. In Table 1, we present the testing performance measured by AP. The model trained on UIUC achieves a modest 55.1% AP. The model trained on UCF improves to 68.8%.

This is expected as the training set of UCF is larger (120 images versus 32) and more diverse than UIUC’s.

Train data	Test data	AP
UIUC	SBU	55.1
UCF	SBU	68.8
SBU-Lazy	SBU	77.5
SBU-Recovered	SBU	80.1

Table 1. **Average precision on the SBU test set.** Different training data and annotations are experimented with.

Shadow detection on this new domain of images is challenging for models trained on existing datasets. However, a model trained on lazy labels from images of a similar domain, achieves AP of 77.5%, which is higher than the APs obtained by models trained on the UCF and UIUC datasets. Furthermore, if we apply the proposed label recovery method, and train with recovered labels, performance increases to 80.1%. Qualitative results shown in Figure 5.

Figure 4 shows some examples of recovered labels. Initial shadow masks from lazy annotation are overlaid in blue. They correspond to the main image shadows. Recovered shadow regions are shown with yellow contours. These shadows were missed by the annotator. Missed shadows are often less prominent, e.g., the shadow of the smaller brown column in the top right corner of the top-row image.

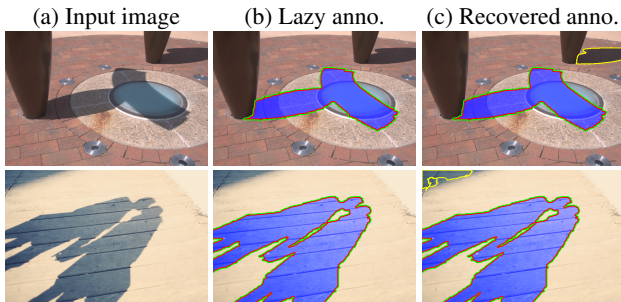


Figure 4. **Example of label recovery.** a) Input image b) Lazy annotation shadow mask overlaid in blue, outer contour in green, inner contour in red. c) Recovered regions with flipped shadow label are shown with yellow contours.

## 5.2. Noisy labels and benefits of label recovery

For the set of controlled experiments that follow, we re-labeled the UCF and UIUC training sets using lazy annotation. We train a classifier using the lazy labels and measure classification performance in the respective test sets. Table 2 shows the classification performance in terms of AP. For UIUC, the performance of the model trained on lazy labels deteriorates by 10% compared to training with the original labels (79.5% vs 88.5%). However, the model trained on recovered lazy labels achieves comparable AP of 87.2%. Qualitative results shown in Figure 6. For UCF, train-

ing with lazy labels is slightly worse than training with original labels, 73.5% versus 74.5%. Interestingly, label recovery improves the classification performance to 75.6%, outperforming the model trained on the original labels. These experiments suggest that we can achieve similar results with recovered lazy labels as with carefully annotated labels, at a small fraction of the annotation effort.

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

Table 2. **Classification performance on UIUC and UCF test sets.** Comparison of AP achieved by the region classifier trained with original carefully annotated labels (Original), lazily annotated labels (Lazy), and recovered lazy labels (Recovered).

## 5.3. Analysis of label noise

It is well established that label noise degrades classification performance. To gain more insight on the effects of label noise in the shadow detection task we perform a series of controlled experiments and deliberately corrupt the labels of the training data of UIUC and UCF. We train the classifier with different levels of label noise and then test the resulting models. Hereafter, we refer to shadow labels as positives and non shadow labels as negatives.

We first focus on corrupted positive labels. We randomly flip negative training samples to positive thus polluting the positive label class (dirty positives). In these experiments, we fix the ratio of dirty negatives and then increasingly pollute the positive class. We measure the classification performance for: (i) A completely clean negative class, (ii) a class with 10% dirty negatives and (iii) a class with 20% dirty negatives. In each of the 3 experiments, we progressively increase the level of label noise for the positive class (“dirty positives”). Results are in Figure 7. We then perform the same analysis for noise in the negative labels, with a set of 3 symmetric experiments where the positive and negative sets are reversed. Results are in Figure 8.

The classifier performance is more sensitive to the amount of dirty negatives than dirty positives. This is due to the smaller size of the positive class, so the effect of erroneously flipping positives is more pronounced. We observe that the effect of dirty negatives is more pronounced in the UIUC dataset. This explains why performance dropped so much when using “lazy” annotation and had a large improvement after label recovery.

In Figure 9, we show the performance of our label recovery approach as the levels of dirty positives and dirty

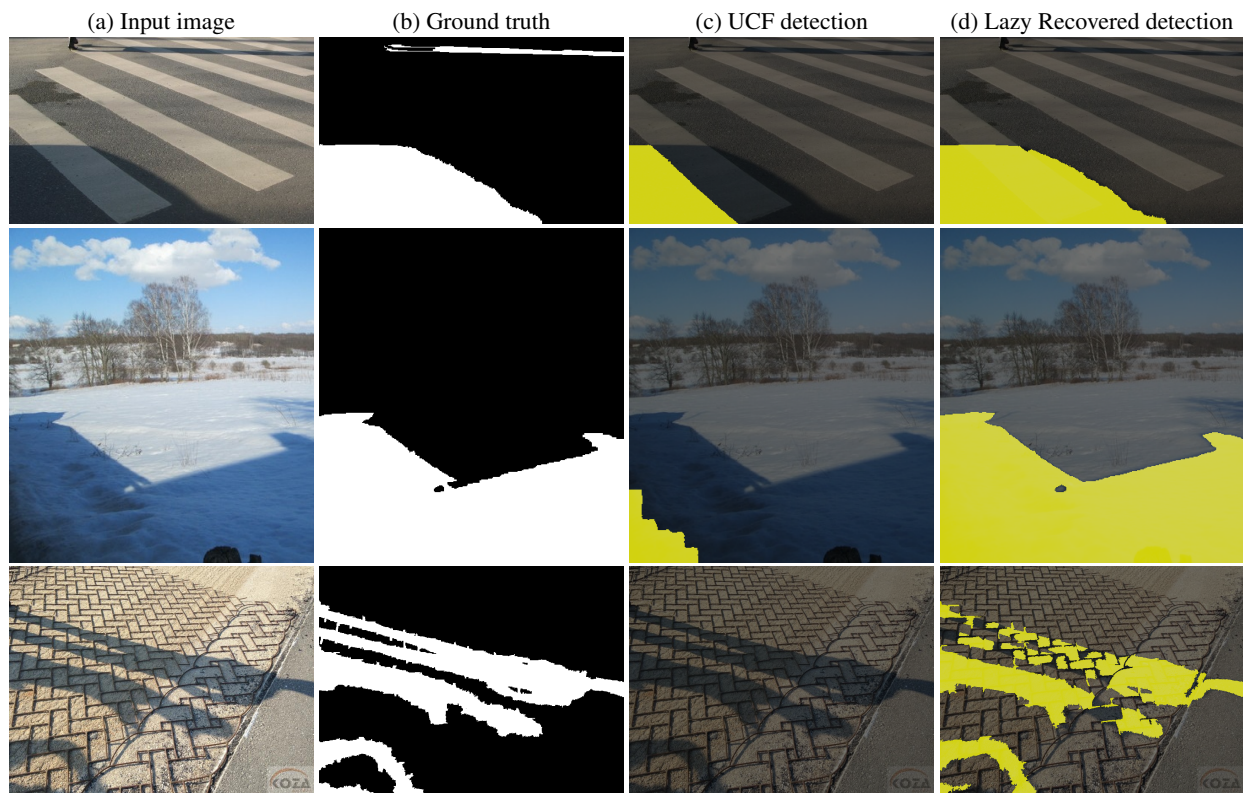


Figure 5. **Shadow detection on a new domain.** Qualitative comparison of detection results for a model trained on UCF with good labels and a model trained on SBU training set with lazy labels that have been recovered. a) Input image. b) Ground truth mask. c) Detection results from classifier trained on UCF overlaid in yellow. d) Detection results from classifier trained with lazy recovered labels overlaid in yellow.

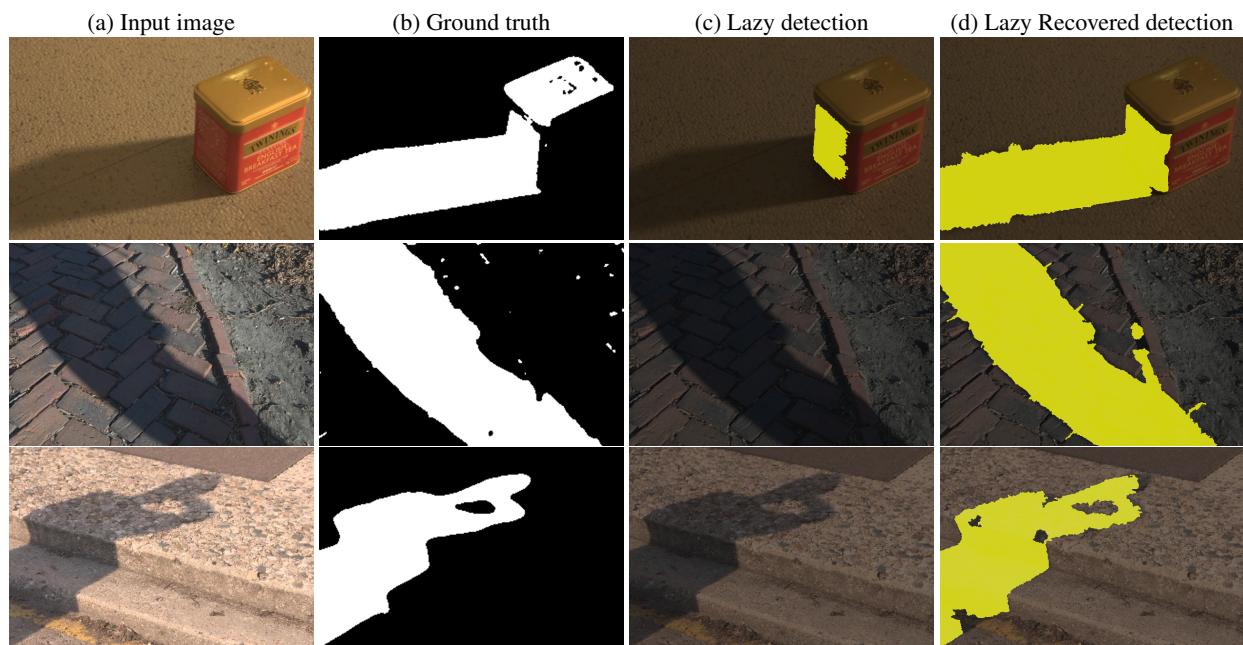


Figure 6. **Shadow detection comparison between models trained with lazy labels and recovered labels on UIUC.** a) Input image. b) Provided manual shadow mask. c) Detection results from model trained on lazy labels overlaid in yellow. d) Detection results from model trained on recovered lazy labels overlaid in yellow.

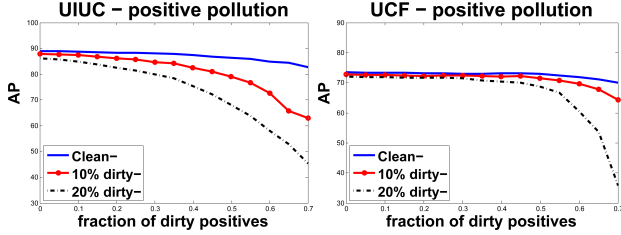


Figure 7. **Effects of positive pollution.** Classification performance at fixed levels of negative noise. Curves show the performance as a function of the proportion of dirty positives, at a fixed level of dirty negatives, either 0%, 10%, or 20%.

negatives vary. The subfigure on the left shows the results of label recovery for increasingly higher levels of positive noise (at fixed 10% negative label noise). The label recovery method improves classification performance when there is up to 30% of positive label noise. Similarly, the right plot shows the results of our label recovery for increasingly higher levels of negative noise (at fixed 10% positive label noise). The label recovery method improves classification performance when the level of noise in negative labels is 50% or less. In both cases label recovery is robust up to significant levels of label noise in the training set.

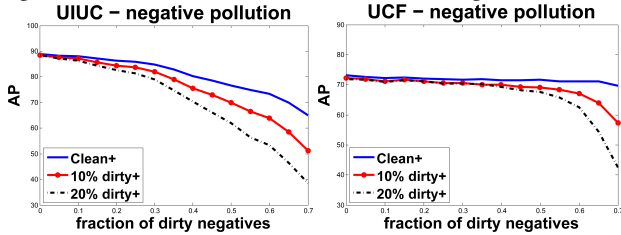


Figure 8. **Effects of negative pollution.** Classification performance at fixed levels of positive noise. Curves show the classification performance as a function of the proportion of dirty negatives, at a fixed level of dirty positives, either 0%, 10%, or 20%.

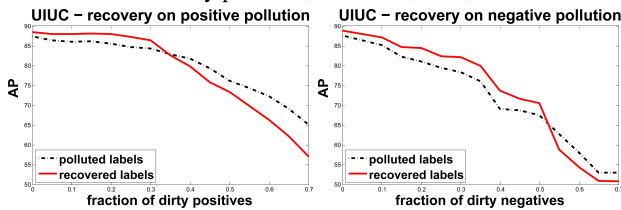


Figure 9. **Label recovery on polluted labels.** Left: Classification performance at fixed 10% of dirty negatives and increasing levels of positive pollution (x-axis). The dash black line is performance without label recovery. The solid red line shows the performance of label recovery. The label recovery improves the performance when the level of positive noise is 30% or less. **Right:** Symmetric analysis as the left subfigure. Classification performance at fixed 10% of dirty positives. The label recovery improves the performance when the level of negative noise is 50% or less.

#### 5.4. Comparison to other noise-tolerant methods

We implemented the noise-tolerant C-SVM method [19]. On the noisy UIUC and UCF datasets, it achieved

an average precision of 81.5 and 74.3, respectively. These are significantly worse than results of our method (87.2 and 75.6 respectively). We also tested our method for noisy labels on the UCI datasets used in [19], and found that it is effective in leveraging noisy labels, as reported in Table 3, suggesting a more general applicability of our method.

Data	$\rho_+$	$\rho_-$	$l_{\log}$ [19]	svm[19]	[16]	[4]	[24]	Ours
Brea.	.2	.2	70.1	67.9	69.3	64.9	69.4	<b>74.5</b>
	.3	.1	70.1	67.8	67.8	65.7	66.3	<b>74.0</b>
	.4	.4	67.8	67.8	67.1	56.5	54.2	<b>72.3</b>
Diab.	.2	.2	<b>76.0</b>	66.4	69.5	73.2	75.0	<b>76.8</b>
	.3	.1	<b>75.5</b>	66.4	65.9	<b>74.7</b>	67.7	<b>75.1</b>
	.4	.4	65.9	65.9	65.4	<b>71.1</b>	62.8	67.3
Thyr.	.2	.2	87.8	94.3	<b>96.2</b>	78.5	84.0	93.8
	.3	.1	80.3	92.5	86.9	87.8	83.1	<b>95.6</b>
	.4	.4	83.1	66.3	71.0	86.0	58.0	<b>88.9</b>
Germ.	.2	.2	71.8	68.4	63.8	67.8	62.8	<b>75.8</b>
	.3	.1	71.4	68.4	67.8	67.8	67.4	<b>77.7</b>
	.4	.4	67.2	68.4	67.8	54.8	59.8	<b>72.6</b>
Heart	.2	.2	<b>83.0</b>	61.5	69.6	<b>83.0</b>	72.9	80.7
	.3	.1	<b>84.4</b>	57.0	62.2	81.5	79.3	79.7
	.4	.4	57.0	54.8	53.3	52.6	68.2	<b>70.3</b>
Image	.2	.2	82.5	<b>92.0</b>	<b>92.9</b>	77.8	65.3	91.3
	.3	.1	82.6	<b>89.3</b>	<b>89.6</b>	79.4	70.7	83.9
	.4	.4	63.5	63.5	73.2	69.6	64.7	<b>81.9</b>

Table 3. **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.

## 6. 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 have also shown the advantage of quickly annotating an appropriate dataset when faced with the task of detecting shadows in a new image domain. We will create large scale shadow datasets with relatively good annotation. We can also adapt the method to combine datasets collected under different annotation methodologies. Such datasets would contribute to the progress of shadow detection and scene understanding. We will explore generalizing label recovery to other domains.

**Acknowledgments.** Partially supported by NSF IIS-1161876, FRA DTFR5315C00011, the Stony Brook SensonCAT, the Sub-sample project from DIGITEO Institute, France.



## 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 Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2281, 2012. 4
- [2] B. Biggio, B. Nelson, and P. Laskov. Support vector machines under adversarial label noise. In *ACML*, pages 97–112, 2011. 3
- [3] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings of the International Conference on Computer Vision*, 2001. 4
- [4] K. Crammer and D. D. Lee. Learning via gaussian herding. In *NIPS*, 2010. 8
- [5] G. Finlayson, S. Hordley, C. Lu, and M. Drew. On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006. 1
- [6] G. Finlayson, M. Drew, and C. Lu. Entropy minimization for shadow removal. *International Journal of Computer Vision*, 85:35–57, 2009. 1
- [7] B. Frenay and M. Verleysen. Classification in the presence of label noise: A survey. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(5):845–869, May 2014. 1, 3
- [8] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1, 3, 4
- [9] R. Guo, Q. Dai, and D. Hoiem. Single-image shadow detection and removal using paired regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1, 2, 5
- [10] R. Guo, Q. Dai, and D. Hoiem. Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2956–2967, 2012. 1, 2
- [11] M. Hoai. Regularized max pooling for image categorization. In *Proceedings of the British Machine Vision Conference*, 2014. 3
- [12] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *Proceedings of the Asian Conference on Computer Vision*, 2014. 3
- [13] X. Huang, G. Hua, J. Tumblin, and L. Williams. What characterizes a shadow boundary under the sun and sky? In *Proceedings of the International Conference on Computer Vision*, 2011. 1
- [14] X. Jiang, A. Schofield, and J. Wyatt. Shadow detection based on colour segmentation and estimated illumination. In *bmvc*, pages 87.1–87.11, 2011. 1
- [15] S. Khan, M. Bennamoun, F. Sohel, and T. Togneri. Automatic feature learning for robust shadow detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1
- [16] R. Khardon and G. Wachman. Noise tolerant variants of the perceptron algorithm. *Journal of Machine Learning Research*, 8:227–248, May 2007. 8
- [17] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *Proceedings of the European Conference on Computer Vision*, 2010. 1
- [18] 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 *eccv*, 2014. 5
- [19] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *NIPS*, 2013. 3, 8
- [20] A. Panagopoulos, D. Samaras, and N. Paragios. Robust shadow and illumination estimation using a mixture model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1
- [21] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios. Simultaneous cast shadows, illumination and geometry inference using hypergraphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):437–449, 2013. 1
- [22] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the International Conference on Machine Learning*, 1998. 3
- [23] L. Shen, T. W. Chua, and K. Leman. Shadow optimization from structured deep edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1
- [24] G. Stempfel and L. Ralaivola. Learning kernel perceptrons on noisy data using random projections. *Algorithmic Learning Theory*, pages 328–342, 2007. 3, 8
- [25] G. Stempfel and L. Ralaivola. Learning svms from sloppily labeled data. In C. Alippi, M. Polycarpou, C. Panayiotou, and G. Ellinas, editors, *Artificial Neural Networks ICANN 2009*, volume 5768 of *Lecture Notes in Computer Science*, pages 884–893. Springer Berlin Heidelberg, 2009. 3
- [26] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999. 3
- [27] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proceedings of the European Conference on Computer Vision*, 2002. 5

- [28] T. F. Y. Vicente, C.-P. Yu, and D. Samaras. Single image shadow detection using multiple cues in a supermodular MRF. In *Proceedings of the British Machine Vision Conference*, 2013. 1, 4
- [29] T. Y. Vicente, M. Hoai, and D. Samaras. Leave-one-out kernel optimization for shadow detection. In *Proceedings of the International Conference on Computer Vision*, 2015. 1, 3
- [30] Z. Wei and M. Hoai. Region ranking svms for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3
- [31] J. Zhu, K. Samuel, S. Masood, and M. Tappen. Learning to recognize shadows in monochromatic natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1, 5
- [32] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004. 1, 3