Automatic Feature Learning for Robust Shadow Detection

S. H. Khan, M. Bennamoun, F. Sohel, R. Togneri
The University of Western Australia
{salman.khan@research.,mohammed.bennamoun@,ferdous.sohel@,roberto.togneri@}uwa.edu.au

Abstract

We present a practical framework to automatically detect shadows in real world scenes from a single photograph. Previous works on shadow detection put a lot of effort in designing shadow variant and invariant hand-crafted features. In contrast, our framework automatically learns the most relevant features in a supervised manner using multiple convolutional deep neural networks (ConvNets). The 7-layer network architecture of each ConvNet consists of alternating convolution and sub-sampling layers. The proposed framework learns features at the super-pixel level and along the object boundaries. In both cases, features are extracted using a context aware window centered at interest points. The predicted posteriors based on the learned features are fed to a conditional random field model to generate smooth shadow contours. Our proposed framework consistently performed better than the state-of-the-art on all major shadow databases collected under a variety of conditions.

1. Introduction

Shadows provide useful clues of the scene characteristics which can help in visual scene understanding. As early as the time of Da Vinci, the properties of shadows were well studied [7]. Recently, shadows have been used for tasks related to object shape [17, 18], size, movement [14], number of light sources and illumination conditions [23]. Shadows have a particular practical importance in augmented reality applications, where the illumination conditions in a scene can be used to seamlessly render virtual objects and their casted shadows. In digital photography, information about shadows and their removal can help to improve the visual quality of photographs. Beside the above mentioned assistive roles, shadows can also cause complications in many fundamental computer vision tasks. They can degrade the performance of object recognition, stereo, shape reconstruction, image segmentation and scene analysis. Shadows are also a serious concern for aerial imaging and object tracking in video sequences [21].

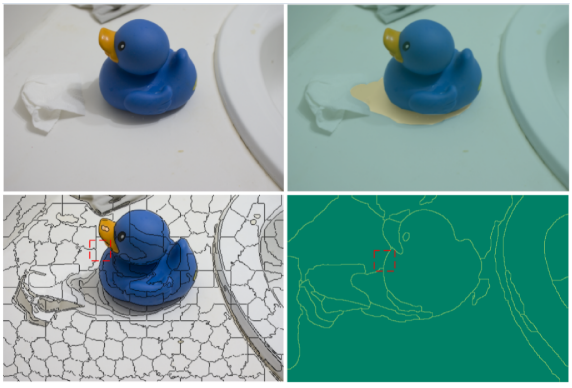


Figure 1: ***Top:*** A photograph with its shadow map on the right. ***Bottom:*** The extracted local information from homogeneous regions (*left*) and along the boundaries (*right*) is used in our approach to detect shadows. (*Best viewed in color*)

Despite the ambiguities generated by shadows, the Human Visual System (HVS) does not face any real difficulty in filtering out the degradations caused by shadows. We need to equip machines with these same visual comprehension abilities. Inspired by the hierarchical architecture of the human visual cortex, many deep representation learning architectures have been proposed in the last decade. We draw our motivation from the recent successes of these deep learning methods in many computer vision tasks where learned features out-performed hand-crafted features [6, 10]. On that basis, we propose to use multiple convolutional neural networks (ConvNets) to learn useful feature representations for the task of shadow detection. ConvNets are biologically inspired deep network architectures based on Hubel and Wiesel's [11] work on the cat's primary visual cortex. To the best of our knowledge, we are the first to use *'learned features'* in the context of shadow detection, as opposed to the common carefully designed and hand-crafted features.

Our proposed approach combines local information at image patches with the local information across boundaries (Fig. 1). Since the regions and the boundaries exhibit different types of features, we split our framework into two respective portions. Separate ConvNets are consequently

trained for patches extracted around the scene boundaries and the super-pixels. Predictions made by the ConvNets are local and we therefore need to exploit the higher level interactions between the neighboring pixels. For this purpose, we incorporate local beliefs in a Conditional Random Field (CRF) model which enforces the labeling consistency over the nodes of a grid graph defined on an image (Sec. 3). This removes isolated and spurious labeling outcomes and encourages neighboring pixels to adopt the same label.

2. Background and Contributions

One of the most popular methods to detect shadows is to use a variety of shadow variant and invariant cues to capture the statistical and deterministic characteristics of shadows [28, 15, 12, 9, 22]. The extracted features model the chromatic, textural [28, 15, 9, 22] and illumination [12, 20] properties of shadows to determine the illumination conditions in the scene. Some works give more importance to features computed across image boundaries, such as intensity and color ratios across boundaries and the computation of texton features on both sides of the edges [27, 15]. Although these feature representations are useful, they are based on assumptions that may not hold true in all cases. As an example, chromatic cues assume that the texture of image regions remains the same across shadow boundaries and only the illumination is different. This approach fails when the image regions under shadows are barely visible. Moreover, all of these methods involve a considerable effort in the design of hand-crafted features for shadow detection and their selection (e.g., the use of ensemble learning methods to rank the best features [28, 15]). Our data-driven framework is unique, in the sense that instead of focusing our efforts on the careful design of hand-crafted features, we propose to use deep feature learning methods to *learn the most relevant features* for shadow detection.

Owing to the challenging nature of the shadow detection problem, many simplistic assumptions are commonly adopted. Previous works made assumptions related to the illumination sources [23], the geometry of the objects casting shadows and the material properties of the surfaces on which shadows are cast. For example, [22] considers object cast shadows while [15, 24] only detect shadows that lie on the ground. Some methods use synthetically generated training data to detect shadows [19]. Techniques targeted for video surveillance applications take advantage of multiple images [8] or time-lapse sequences [13] to detect shadows. User assistance is also required by some proposed techniques to achieve their attained performances [25, 3]. Methods based on strong assumptions such as known geometry [24] and point light sources casting shadows on a planar lambertian surface [23] are also used in practice. In contrast, our framework makes absolutely *no 'prior assumptions'* about the scene and the shadow properties, the shape of objects, the image capturing conditions and the surrounding environments. Based on this premise, we tested our proposed framework on all of the publicly available databases for shadow detection from single images. These databases contain common real world scenes with artifacts such as noise, compression and color balancing effects. There is an acute need for shadow detection from a single image in such noisy and varied environments.

The key contributions of our work are outlined below:

- A new approach for robust shadow detection combining both regional and across-boundary learned features in a probabilistic framework involving CRFs (Sec. 3).
- Automatic learning of the most relevant feature representations from raw images using multiple ConvNets (Sec. 3.1).
- An extensive quantitative evaluation to prove that the proposed framework is robust, less-constrained and generalisable across different types of scenes (Sec. 4).

3. Proposed Shadow Detection Framework

Given a single color image, we aim to detect and localize shadows precisely at the pixel level (see Fig. 2). If $\mathbf{y}$ denotes the desired binary mask encoding class relationships, we can model the shadow detection problem as a conditional distribution:

$$\mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w}) = \frac{1}{Z(\mathbf{w})} \prod_{i\in\mathcal{V}} \varphi_i^{\mathbf{w}_i}(y_i,\mathbf{x}) \prod_{(i,j)\in\mathcal{E}} \varphi_{ij}^{\mathbf{w}_{ij}}(y_{ij},\mathbf{x}), \quad (1)$$

where, the parameter vector $\mathbf{w}$ includes the weights of the model, the latent variables are represented by $\mathbf{x}$ where $\mathbf{x}_i$ denote the intensity of pixel $i \in \{p_i\}_{1\times N}$ and $Z(\mathbf{w})$ denotes the partition function. The distribution in Eq. 1 can be formulated in terms of the Gibbs energy as follows:

$$\mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w}) = \frac{1}{Z(\mathbf{w})}\exp(-\mathrm{E}(\mathbf{y},\mathbf{x};\mathbf{w})) \quad (2)$$

The energy function is composed of two potentials; the unary potential ψ_i and the pairwise potential ψ_{ij}:

$$\begin{aligned}\mathrm{E}(\mathbf{y},\mathbf{x};\mathbf{w}) &= -\log\mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w}) - \log Z(\mathbf{w}) \\ &= \sum_{i\in\mathcal{V}} \psi_i(y_i,\mathbf{x};\mathbf{w}_i) + \sum_{(i,j)\in\mathcal{E}} \psi_{ij}(y_{ij},\mathbf{x};\mathbf{w}_{ij}) \end{aligned} \quad (3)$$

These energies are related to the potential functions defined in Eq. 1 as: $\varphi_k^{\mathbf{w}_k}(y_k,\mathbf{x}) = \exp(-\psi_k(y_k,\mathbf{x};\mathbf{w}_k))$. with $k \in \{i, ij\}$. The unary potential considers the shadow properties both at the regions and at the boundaries inside the image.

$$\psi_i(y_i,\mathbf{x};\mathbf{w}_i) = \overbrace{\phi_i^r(y_i,\mathbf{x};\mathbf{w}_i^r)}^{\text{region}} + \overbrace{\phi_i^b(y_i,\mathbf{x};\mathbf{w}_i^b)}^{\text{boundary}} \quad (4)$$

Each of the boundary and regional potentials is defined in terms of probability estimates from the two separate ConvNets,

$$\begin{aligned}\phi_i^r(y_i,\mathbf{x};\mathbf{w}_i^r) &= -\mathbf{w}_i^r \log\mathcal{P}_{\text{cnn1}}(y_i|\mathbf{x}_\text{r}) \\ \phi_i^b(y_i,\mathbf{x};\mathbf{w}_i^b) &= -\mathbf{w}_i^b \log\mathcal{P}_{\text{cnn2}}(y_i|\mathbf{x}_\text{b})\end{aligned} \quad (5)$$

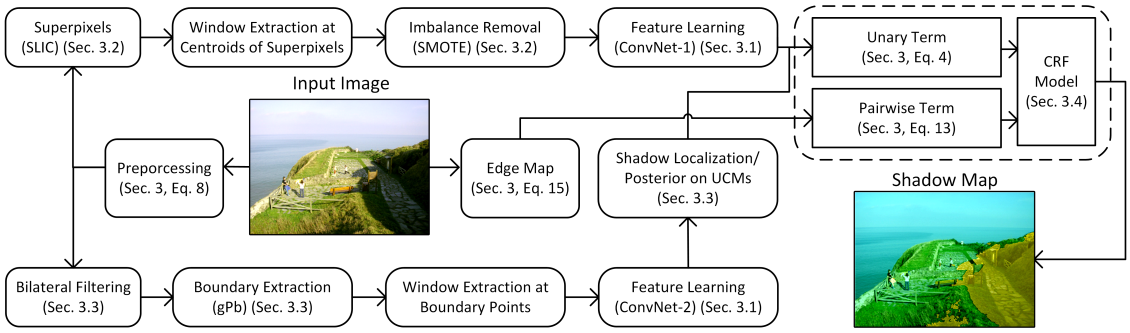


Figure 2: The proposed shadow detection framework. (*Best viewed in color*)

This is logical because the features to be estimated at the boundaries are likely to be different from the ones estimated inside the shadowed regions. Therefore, we train two separate ConvNets, one for the regional potentials and the other for the boundary potentials. ConvNets operate on equi-sized windows, so it is required to extract patches around the desired interest points using a windowing operation ($\mathcal{W}^{i,j}(\cdot)$ in Eq. 6). For the case of regional potentials, we extract super-pixels by clustering the homogeneous pixels ($\mathcal{F}_{\text{slic}}(\cdot)$ in Eq. 6). Although any super-pixel extraction method can be used, we opted to use a recently proposed technique called SLIC [1], due to its efficiency. Afterwards, a patch ($\mathcal{I}^r$) is extracted by centering a $\tau_s \times \tau_s$ window at the centroid of each superpixel. For the spatial location (i, j) of a centroid, this operation can be represented as:

$$\mathcal{I}^r(i,j) = \mathcal{W}^{i,j}(\mathcal{F}_{\text{slic}}(\mathbf{x}), \tau_s). \tag{6}$$

Similarly for boundary potentials, we extract boundaries inside an image using the gPb technique ($\mathcal{F}_{\text{gPb}}(\cdot)$ in Eq. 7) [2]. We traverse along each boundary with a stride λ_b and extract a $\tau_s \times \tau_s$ patch at each step to incorporate local context. Therefore, similar to Eq. 6, we have:

$$\mathcal{I}^b(i,j) = \mathcal{W}^{i,j}(\mathcal{F}_{\text{gPb}}(\mathbf{x}), \tau_s). \tag{7}$$

Using ConvNets, we want to model the distributions: $\mathcal{P}_{\text{cnn1}}(y_i|\mathcal{I}^r(i,j))$ and $\mathcal{P}_{\text{cnn2}}(y_i|\mathcal{I}^b(i,j))$ as:

$$\mathcal{P}_{\text{cnn}}(y_i|\mathcal{I}(i,j)) = \mathcal{C}(\theta(\mathcal{I}(i,j))), \tag{8}$$

where $\theta(\cdot)$ is the pre-processor, $\mathcal{I}$ can be either a boundary or a super-pixel patch and $\mathcal{C}(\cdot)$ is a ConvNet with five hidden layers. The distribution in Eq. 8 is related to those in Eq. 5, since $\mathbf{x}_{\text{r}} = \{\mathcal{I}^r(i,j)\}_{1\times|\mathcal{F}_{\text{slic}}(\mathbf{x})|}$ and $\mathbf{x}_{\text{b}} = \{\mathcal{I}^b(i,j)\}_{1\times\frac{|\mathcal{F}_{\text{gPb}}(\mathbf{x})|}{\lambda_b}}$, where $|.|$ is the cardinality operator. The response of each convolution node is given by:

$$\mathbf{a}_n^l = \sigma(\sum_{\forall m}(\mathbf{a}_m^{l-1} * \mathbf{k}_{m,n}^l) + b_n^l) \tag{9}$$

where, $\mathbf{a}_n^l$ and $\mathbf{a}_n^{l-1}$ are the feature maps of the current layer l and the previous layer $l-1$ respectively, $\mathbf{k}$ is the convolution kernel and indices (m, n) show the mapping from m^{th} feature map of the previous layer to the n^{th} feature map in the current layer. The $\sigma(\cdot)$ represents the element-wise nonlinear activation function and b denotes the bias node. The response of each sub-sampling node is given by:

$$\mathbf{a}_n^l = \sigma(k_n^l \times \frac{1}{S^2}\sum_{S\times S}\mathbf{a}_n^{l-1} + b_n^l) \tag{10}$$

where, k_n^l is the weight and $S \times S$ is the size of the patch on which the values are averaged. The response of a neuron in the output layer is given by:

$$\mathbf{a}_n^{out} = \sigma(\sum_{\forall m}(\mathbf{a}_m^{out-1} \times k_{m,n}^{out}) + b_n^{out}) \tag{11}$$

The posterior distribution on the binary output variables ($n \in \{shdw, n\text{-}shdw\}$ in Eq. 11) is defined as:

$$\mathcal{P}_{\text{cnn}}(y_i|\mathcal{I}(i,j)) = [\mathbf{a}_{shdw}^{out} \quad \mathbf{a}_{n\text{-}shdw}^{out}]. \tag{12}$$

Here, for the case of the regions, the posteriors predicted by the ConvNet are assigned to each super pixel in an image. However, for the boundaries, we first localize the probable shadow location and then average the predicted probabilities over each contour (See Sec. 3.3).

The pairwise potential in Eq. 3 is defined as a combination of the class transition potential ϕ_{p_1} and the spatial transition potential ϕ_{p_2}:

$$\psi_{ij}(y_{ij}, \mathbf{x}; \mathbf{w}_{ij}) = \mathbf{w}_{ij}\phi_{p_1}(y_i, y_j)\phi_{p_2}(\mathbf{x}). \tag{13}$$

The class transition potential is defined as an Ising prior:

$$\phi_{p_1}(y_i, y_j) = \alpha\mathbf{1}_{y_i \neq y_j} = \begin{cases} 0 & \text{if } y_i = y_j \\ \alpha & \text{otherwise} \end{cases} \tag{14}$$

The spatial transition potential captures the differences in the adjacent pixel intensities:

$$\phi_{p_2}(\mathbf{x}) = [\exp(-\frac{\|x_i - x_j\|^2}{\beta_x\langle\|x_i - x_j\|^2\rangle})] \tag{15}$$

where, $\langle\cdot\rangle$ denotes the average contrast in an image. The parameters α and β_x were derived from a cross validation on each database.

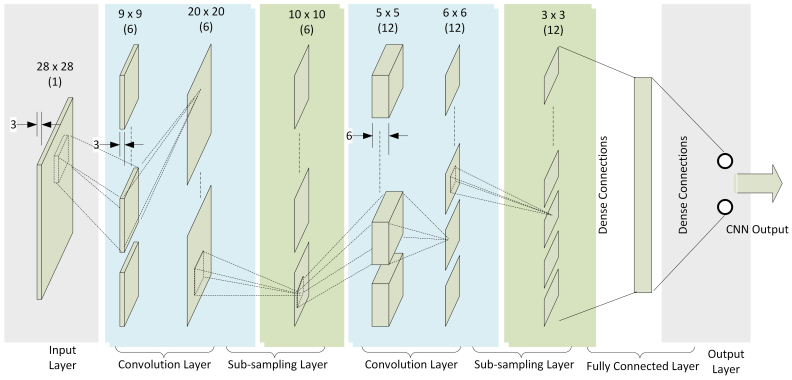
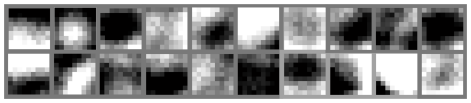
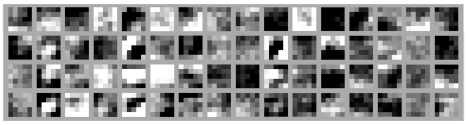


Figure 3: ConvNet architecture used for automatic feature learning



(a) Examples of 9×9 learned kernels for bottom (2^{nd} from left in Fig. 3) convolution layer



(b) Examples of 5×5 learned kernels for top (4^{th} from left in Fig. 3) convolution layer

Figure 4: Visualization of the learned kernels

When making an inference, the most likely labeling is found by using the Maximum a Posteriori (MAP) estimate ($\mathbf{y}^*$) upon a set of random variables $\mathbf{y} \in \mathcal{L}^N$. This estimate turns out to be an energy minimization problem since the partition function $Z(\mathbf{w})$ does not depend on $\mathbf{y}$:

$$\mathbf{y}^* = \underset{\mathbf{y} \in \mathcal{L}^N}{\operatorname{argmax}} \mathcal{P}(\mathbf{y}|\mathbf{x};\mathbf{w}) = \underset{\mathbf{y} \in \mathcal{L}^N}{\operatorname{argmin}} \mathrm{E}(\mathbf{y},\mathbf{x};\mathbf{w}) \tag{16}$$

The parameters ($\mathbf{w}$) in Eq. 16 are learnt using a max-margin criterion, details of which are given in [26]. In the next section, we outline the details of the ConvNet architecture.

3.1. Feature Learning with ConvNets

We employ multiple ConvNets for feature learning along the boundaries and at the super-pixel level. The same ConvNet architecture was used for feature learning at each of these levels (see Fig. 3). This consists of alternating convolution and sub-sampling layers together with a fully connected layer located just before the output layer. This layered structure enables ConvNets to learn multilevel hierarchies of features. The network architecture takes an RGB patch as an input and processes it to give a posterior distribution over binary classes. Each neuron output is modeled as a nonlinear function ($\sigma(\cdot)$) of its input, which is defined by a logistic sigmoid function, $\sigma(x) = (1 + e^{-x})^{-1}$. The convolution layers in ConvNets consist of filter banks which are convolved with the input feature maps (Eq. 9). The sub-sampling layers combine the outputs from the neighboring groups of neurons in the same kernel map (Eq. 10). We define the stride of this pooling operation to be equal to the pooling neighborhood over which the kernel elements are averaged. The pooling operation performed by the sub-sampling layers helps in learning invariant feature representations.

A fully connected layer appears just before the output layer. It has dense connections with the previous layer i.e., its input is the set of all feature maps of the final sub-sampling layer. The fully connected layer works as a traditional MLP with one hidden layer followed by a logistic regression output layer which provides a distribution over the classes. In our ConvNet architecture, the kernels of each convolution layer are connected with all the kernel maps of the previous layer. However, each kernel map in the sub-sampling layer is only connected to the corresponding kernel map of the previous layer. For the two convolution layers, we use 6 and 12 kernels of size 9×9 and 5×5 respectively (Fig. 4). A unit pixel stride is chosen for the center of the receptive fields.

The ConvNets operate on equi-sized patches extracted around the super-pixels and the boundaries (Eq. 6, 7). Before feeding the extracted patches to each ConvNet, the data is zero-centered and normalized (θ operator in Eq. 8). We empirically found that pixel decorrelation methods (e.g., whitening) do not help in the shadow detection task. After pre-processing, a supervised training of the ConvNet is performed using online learning (stochastic gradient descent), which showed to be more efficient compared to batch learning. During the training process, the gradients are computed using the back-propagation method and the cross entropy loss function is minimized [16]. We set the training parameters (such as momentum, weight decay) by cross validation. The training samples are shuffled randomly before training since the network can learn faster from unexpected samples. The weights of the ConvNet were initialized using randomly drawn samples from a Gaussian distribution of zero mean and a variance that is inversely proportional to the *fan-in* measure of neurons.

The number of epochs during the training of ConvNets is set by an early stopping criterion. For this purpose a small validation set is used to evaluate the trained network after every epoch. The training process is halted once the performance on the validation set does not increase in δ successive steps ($\delta = 5$ in our case). The trained network with the best performance on the validation set is then used for the actual testing. The initial learning rate is heuristically chosen by selecting the largest rate which resulted in the convergence of the training error.

3.2. Region Extraction and Imbalance Learning

We generate over-segmented images using the *Simple Linear Iterative Clustering* (SLIC) algorithm [1]. In this way, homogeneous regions are clustered in an unsupervised manner to form super-pixels. The centroid of each super-pixel is found and a fixed size window is extracted around the interest points. However, shadows have a limited representation in natural scenes which results in a skewed class distribution of the training set. The ratios between shadowed versus non-shadowed pixels are approximately 1:6, 1:4 and 1:4 for the case of UCF, UIUC and CMU databases, respectively (Sec. 4.1). We address this class imbalance problem at the data level using the Synthetic Minority Oversampling Technique (SMOTE) [5]. It is a bootstrapping technique which synthetically generates training samples to increase the representation of the minority class. It interpolates between closely lying training samples to generate synthetic data: $\mathbf{z}' = \mathbf{z}_i + \omega(\mathbf{z}_i^k - \mathbf{z}_i)$, where, $\mathbf{z}_i^k$ is a randomly selected k nearest neighbor of training sample $\mathbf{z}_i$ and $\omega \in [0, 1]$ is a random weight. For ConvNets, this artificial enlargement of the dataset through the application of label preserving transformations proved to be an easy and elegant way to reduce the overfitting problem (e.g., in [6]).

3.3. Boundary Detection and Shadow Localization

An edge aware smoothness filter (Bilateral filter[1]) was applied before boundary extraction to enhance the edges. Next, the gPb boundary detector [2] was used to find the significant boundaries in an image. We extracted windows along the boundaries after every λ_b boundary points[2]. The overlapping boundary patches are then fed to a ConvNet for training. The trained ConvNet differentiates between the shadow and reflectance edges and predicts the class belonging probabilities based on the trained weights. Next, the probable shadow portion is localized using the dominant shadow properties and the appropriate posterior predicted by the ConvNet is assigned to the localized region. We also calculate the Ultra-metric Contour Maps (UCM) and thus the hierarchical segmentation regions. The assigned probabilities are averaged inside each segmented region to end up with a uniform posterior distribution inside each homogeneous patch (see Algorithm 1).

Algorithm 1 Boundary Detection and Shadow Localization

Input: $\mathcal{I}$: Image with shadow
Output: $\mathcal{O}$: 3D Matrix of posterior probabilities
Initialize: $\gamma_{th} \leftarrow 0.2$; $\lambda_b \leftarrow \tau_s/4$; Patch dictionary, $\mathcal{T} \leftarrow \phi$; Posteriors predicted by ConvNet operating on boundaries, $\mathcal{P} \leftarrow \phi$; $idx = 0$
Calculate boundaries: $\mathbf{B} \leftarrow \mathcal{F}_{\text{gPb}}(\mathcal{I})$
Get hierarchical segmentations: $\mathbf{U} \leftarrow \mathcal{F}_{\text{ucm}}(\mathbf{B})$
$\mathbf{B}_{th} \leftarrow (\mathbf{B} >= \gamma_{th})$
All unique boundary strengths: $\mathbf{b} \leftarrow \{B_n^{ij}\}_{n \in \{1 \dots N\}}$
for $n \leftarrow 1 \dots N$ **do**
  $idx \leftarrow idx + 1$; $M \leftarrow |\mathbf{B}_{th} == \mathbf{b}_n|/\lambda_b$
  for $m \leftarrow 1 \dots M$ **do**
    $\{i, j\} \leftarrow loc([\mathbf{B}_{th} == \mathbf{b}_n]_{m \times \lambda_b + 1})$
    Extract window at location $\{i, j\}$ corresponding to n^{th} boundary and m^{th} point; $\mathcal{T}[idx] \leftarrow \mathcal{W}(\mathcal{I}(i, j))$
    $\mathcal{P}[idx] \leftarrow \mathcal{F}_{\text{CNN2}}(\mathcal{T}[idx])$
    if $\mathcal{P}_{idx} \in \mathcal{S}$ (shadow class) **then**
      Convert to grayscale; $I \leftarrow \mathcal{F}_{\text{grayscale}}(\mathcal{T}_{idx})$
      Calculate edges; $I \leftarrow \mathcal{F}_{\text{canny-edge}}(I)$
      Crop the patch to remove the unit width border
        $I_{\text{crop}} \leftarrow I[2 : (end - 1)][2 : (end - 1)]$
      Diagonal-fill morphological operator to remove diagonal connectivity of background; $I_{crop} \leftarrow \mathcal{F}_{diag}(I_{crop})$
      Convert to a binary image; $I_{\text{binary}}^{i,j} \leftarrow (I_{\text{crop}}^{i,j} <= 0.5)$
      Label connected components in I_{binary} using 8-neighborhood system, $\mathcal{L}^{i,j} \leftarrow \ell \in [1, L]$
      for each connected region $\mathbf{c} \in \mathcal{L}$ **do**
        Calculate mean intensity; $\bar{\mathbf{c}}_k \leftarrow avg(\mathbf{c}_k)$
      end for
      Locate the minimum mean intensity region in $\mathcal{L}$
      Locate the position of patch in the actual image $\mathcal{I}$
      Assign the probability $\mathcal{P}_{idx}$ to the located region
    end if
  end for
end for
$\mathcal{O} \leftarrow$ Averaged posteriors on each segmented region in $\mathbf{U}$

3.4. Shadows Contour Generation using CRFs

We model our problem in the form of a two-class scene parsing problem where each pixel is labeled either as a shadow or non-shadow. This binary classification problem takes probability estimates from the supervised feature learning algorithm and incorporates them in a CRF model. The CRF model is defined on a grid structured graph topology, where graph nodes correspond to image pixels (Eq. 1). The CRFs prove to be an elegant source of enforcing label consistency and local smoothness over the pixels. However, the size of the training space (labeled images) makes it intractable to compute the gradient of the likelihood and therefore the parameters of the CRF cannot be found by simply maximizing the likelihood of hand labeled shadows. Therefore, we use a max-margin learning algorithm to learn the parameters of our proposed CRF model [26]. Because our proposed energies are sub-modular, we use *graph-cuts* for making efficient inference [4].

[1] with half width of 4, spatial and intensity std of 3 and 0.1 respectively.
[2] the step size is $\lambda_b = \tau_s/4$ to get partially overlapping windows.

4. Experiments and Analysis

4.1. Datasets

UCF Shadow Dataset [28]: It has 355 images together with their manually labeled ground truths. Zhu et al. have used 255/355 images for shadow detection [28].
CMU Shadow Dataset [15]: It consists of 135 consumer grade images with labels for only those shadow edges which lie on the ground plane. Since our algorithm is not restricted to ground shadows, we tested our approach on the

	Methods	UCF Dataset	CMU Dataset	UIUC Dataset
	BDT-BCRF (Zhu et al. [28])	88.70%	–	–
	BDT-CRF-Scene Layout (Lalonde et al. [15])	–	84.80%	–
	Unary SVM-Pairwise (Guo et al. [9])	90.20%	–	89.10%
	Bright Channel-MRF (Panagopoulos et al. [20])	85.90%	–	–
	Illumination Maps-BDT-CRF (Jiang et al. [12])	83.50%	84.98%	–
This paper	ConvNet(Boundary+Region)	89.31%	87.02%	92.31%
This paper	ConvNet(Boundary+Region)-CRF	**90.65%**	**88.79%**	**93.16%**

Table 1: Evaluation of the proposed shadow detection scheme; All performances are reported in terms of pixel-wise accuracies.

more challenging criterion of full shadow detection which required the generation of new ground truths.

UIUC Shadow Dataset [9]: It contains 108 images each of which is paired with its corresponding shadow-free image to generate a ground truth shadow mask.

Test/Train Split: For UCF and UIUC databases, we used the split mentioned in [28, 9]. Since CMU database [15] did not report the split, we therefore used even/odd images for training/testing (following the procedure in [12]).

4.2. Results

We assessed our approach both quantitatively and qualitatively on all the major datasets for single image shadow detection. We demonstrate the success of our shadow detection framework on different types of scenes including beaches, forests, street views, aerial images, road scenes and buildings. The databases also contain shadows under a variety of illumination conditions such as sunny, cloudy and dark environments. For quantitative evaluation, we report the performance of our framework when only the unary term (Eq. 4) was used for shadow detection. Further, we also report the per-pixel accuracy achieved using the CRF model on all the datasets. This means that labels are predicted for every pixel in each test image and are compared with the ground-truth shadow masks. For the UCF and CMU datsets, the initial learning rate of $\eta_0 = 0.1$ was used while for the UIUC dataset we set $\eta_0 = 0.01$. After every 20 epochs the learning rate was decreased by a small factor $\beta = 0.5$ which resulted in a good performance.

Table 1 summarizes and compares the overall results of our approach. It must be noted that the accuracy of Jiang's method [12] (on the CMU database) is given by the Equal Error Rate (EER). All other accuracies represent the highest detection rate achieved, which may not necessarily be an EER. Using the CRF model incorporating ConvNets, we were able to get the best performance on the UCF, CMU and UIUC databases with a respective increase of 0.50%, 4.48% and 4.55% compared to the previous best results. These improvements in accuracies approximately translate into 1.52×10^5, 1.0×10^6 and 6.68×10^5 correctly classified pixels on the UCF, CMU and UIUC databases respectively. Although an accuracy gain of 0.5% on the entire UCF database looks modest, previous best methods [9,28] were only evaluated on a subset of 255/355 images. We report the results on the entire database because the exact subset of the color images is not known. Compared to [12], which is evaluated on the entire database, we achieved a relative accuracy gain of 8.56%. On 255 randomly selected images from UCF database, our method achieved an accuracy of 95.1% with a gain of 5.43% over [9].

Methods/Datasets	Shadows	Non-Shadows
UCF Dataset		
BDT-BCRF [28]	63.9%	93.4%
Unary-Pairwise [9]	73.3%	**93.7%**
Bright Channel-MRF [20]	68.3%	89.4%
ConvNet(Boundary+Region)	72.5%	92.1%
ConvNet(Boundary+Region)-CRF	**78.0%**	92.6%
CMU Dataset		
BDT-CRF-Scene Layout [15]	73.1%	**96.4%**
ConvNet(Boundary+Region)	81.5%	90.5%
ConvNet(Boundary+Region)-CRF	**83.3%**	90.9%
UIUC Dataset		
Unary-Pairwise [9]	71.6%	95.2%
ConvNet(Boundary+Region)	83.6%	94.7%
ConvNet(Boundary+Region)-CRF	**84.7%**	**95.5%**

Table 2: Class-wise accuracies of our proposed framework in comparison with the state-of-the-art techniques. Our approach gives the highest accuracy for the class *'shadows'*.

Table 2 shows the comparison of class-wise accuracies. The true positives are reported as the number of predicted shadow pixels which match with the ground-truth shadow mask. False positives are reported as the number of predicted shadow pixels that lie outside the shadow mask. It is interesting to see that our framework has the highest shadow detection performance on the UCF, CMU and UIUC datasets. The ROC curve comparisons are shown in Fig. 6. These curves represent the performance of the unary detector since we cannot generate ROC curves from the outcome of the CRF model. Our approach achieved the highest AUC measures for all datasets (Fig. 6).

Some representative qualitative results are shown in Fig. 5 and Fig. 7. The proposed framework successfully detects shadows in dark environments (Fig. 5: 1^{st} row, middle image) and distinguishes between dark non-shadow regions and shadow regions (Fig. 5: 2^{nd} row, 2^{nd} and 5^{th} image from left). It performs equally well on satellite images (Fig. 5: last column) and outdoor scenes with street views (Fig.

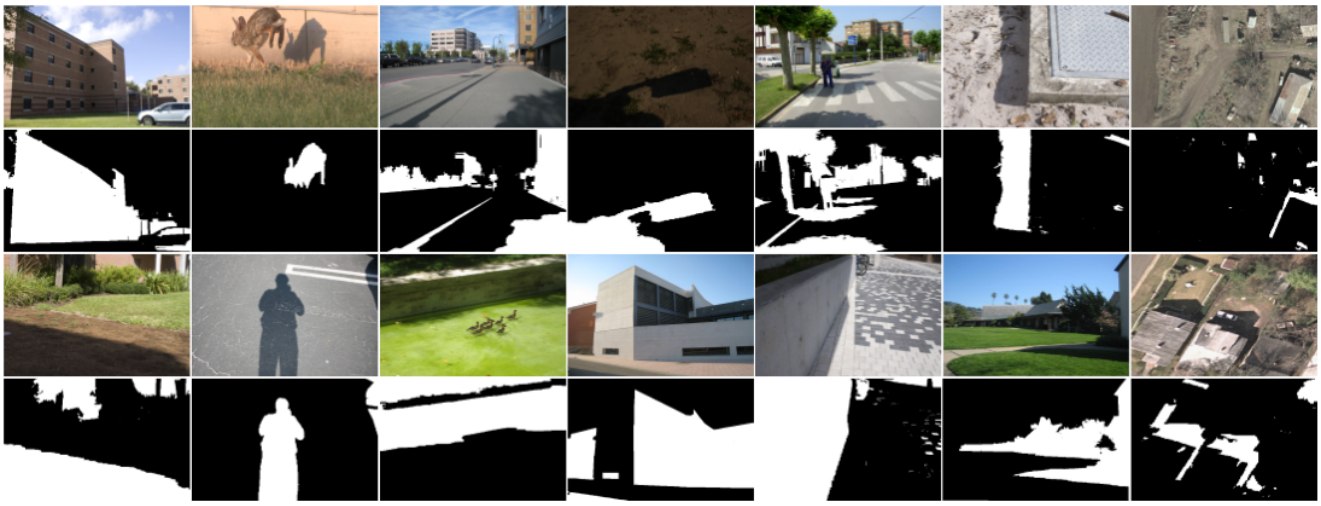
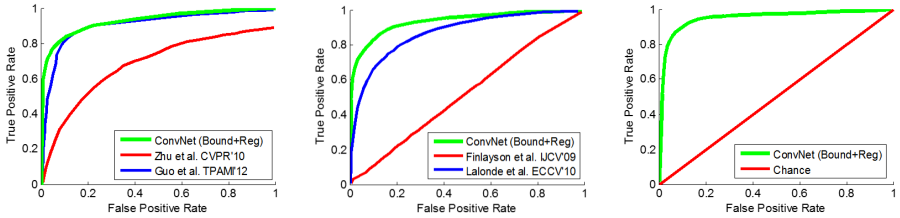


Figure 5: Examples of our results; Images ($1^{st}, 3^{rd}$ row) and shadow masks ($2^{nd}, 4^{th}$ row); Shadows are in white. (*Best viewed in color*)



(a) UCF Shadow Dataset (b) CMU Shadow Dataset (c) UIUC Shadow Dataset

Figure 6: ROC curve comparisons of proposed framework with previous works.

Tested on	Trained on		
	UCF	CMU	UIUC
UCF	–	80.3%	80.5%
CMU	77.7%	–	76.8%
UIUC	82.8%	81.5%	–

Table 3: Results when ConvNets were trained and tested across different datasets

5: 1^{st} row, 3^{rd} and 5^{th} images; 2^{nd} row, middle image), buildings (Fig. 5: 1^{st} column) and shadows of animals and humans (Fig. 5: 2^{nd} column).

4.3. Discussion

The previously proposed methods (e.g., [28, 15]) which used many hand-crafted features, not only require a lot of effort in their design but also require long training times when ensemble learning methods are used for feature selection. As an example, Zhu et al. [28] extracted different shadow variant and invariant features alongside an additional 40 classification results from the Boosted Decision Tree (BDT) for each pixel as their features. Their approach required a huge amount of memory (∼9GB for 125 training images of average size of approximately 480×320). Even after paralleliztion and training on multiple processors, they reported 10 hours of training with 125 images. Lalonde et al. [15] used 48 dimensional feature vectors extracted at each pixel and fed these to a boosted decision tree in a similar manner as [28]. Jiang et al. [12] included illumination features on top of the features used in [15]. Although, enriching the feature set in this way increases performance, it not only takes much more effort to design such features but it also slows down the detection procedure. In contrast, our feature learning procedure is fully automatic and only requires ∼1GB memory and approximately one hour training for each of the UCF, CMU and UIUC databases.

We extensively evaluated our approach on all available databases and our proposed framework turned out to be fairly generic and robust to variations. It achieved the best results on all the single image shadow databases known to us. In contrast, previous techniques were only tested on a portion of database [15], one [28] or at most two databases [9]. Another interesting observation was that the proposed framework performed reasonably well when our ConvNets were trained on one dataset and tested on another dataset. Table 3 summarizes the results of cross-dataset evaluation experiments. These performance levels show that the feature representations learned by the ConvNets across the different datasets were common to a large extent. This observation further supports our claim regarding the generalization ability of the proposed framework.

In our experiments, objects with dark albedo turned out to be a difficult case for shadow detection. Moreover, some ambiguities were caused by the complex self shading patterns created by tree leaves. There were some inconsistencies in the manually labeled ground-truths, in which a shadow mask was sometimes missing for an attached

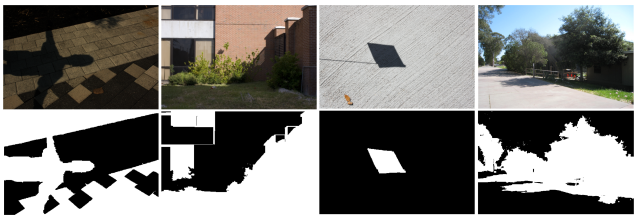


Figure 7: Examples of ambiguous cases: (From *left* to *right*) Our framework misclassified a dark non-shadow region, texture-less black window glass, very thin shadow region and trees due to complex self shading patterns.

shadow. Narrow shadowy regions caused by structures like poles and pipes also proved to be a challenging case for shadow detection. Examples of the above mentioned failure cases are shown in Fig. 7.

5. Conclusion and Future Work

We presented a data-driven approach to learn the most relevant features to detect shadows from a single image. We showed that our framework performs best on a number of databases and it does not depend on the object shape, the environment and the type of scene. In our future work, we will use the proposed shadow detection framework together with the scene geometry (as in [15]) and object properties to reason about high-level scene understanding tasks (as in [19]). The joint training of deep learning architectures over both regions and boundaries will also be explored.

Acknowledgments

This research was supported by the Australian Research Council (ARC) grants DP110102166 and DE120102960.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2282, 2012.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011.

[3] A. Bousseau, S. Paris, and F. Durand. User-assisted intrinsic images. In *TOG*, volume 28, page 130. ACM, 2009.

[4] Y. Boykov and G. Funka-Lea. Graph cuts and efficient nd image segmentation. *IJCV*, 70(2):109–131, 2006.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of AI Research*, 16(1):321–357, 2002.

[6] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649, 2012.

[7] L. Da Vinci. *Notebooks*. Oxford University Press, 2008.

[8] G. Finlayson, C. Fredembach, and M. S. Drew. Detecting illumination in images. In *ICCV*, pages 1–8. IEEE, 2007.

[9] R. Guo, Q. Dai, and D. Hoiem. Paired regions for shadow detection and removal. *TPAMI*, 2012.

[10] M. Hayat, M. Bennamoun, and S. An. Learning non-linear reconstruction models for image set classification. In *CVPR*, 2014.

[11] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106, 1962.

[12] X. Jiang, A. J. Schofield, and J. L. Wyatt. Shadow detection based on colour segmentation and estimated illumination. In *BMVC*, pages 1–11, 2011.

[13] A. J. Joshi and N. P. Papanikolopoulos. Learning to detect moving shadows in dynamic environments. *TPAMI*, 30(11):2055–2063, 2008.

[14] D. Kersten, D. Knill, P. Mamassian, and I. Bülthoff. Illusory motion from shadows. *Nature*, 379(6560):31, 1996.

[15] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *ECCV*, pages 322–335. Springer, 2010.

[16] Y. LeCun, L. Bottou, G. Orr, and K.-R. Mller. Efficient backprop. In *Neural Networks: Tricks of the Trade*, volume 7700 of *LNCS*, pages 9–48. Springer Berlin Heidelberg, 2012.

[17] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi. Illumination normalization with time-dependent intrinsic images. *TPAMI*, 26(10):1336–1347, 2004.

[18] T. Okabe, I. Sato, and Y. Sato. Attached shadow coding: estimating surface normals from shadows under unknown reflectance and lighting conditions. In *ICCV*, pages 1693–1700. IEEE, 2009.

[19] A. Panagopoulos, C. Wang, D. Samaras, and N. Paragios. Illumination estimation and shadow detection through a higher-order model. In *CVPR*, pages 673–680, 2011.

[20] R. Panagopoulos, C. Wang, and D. Samaras. Estimating shadows with the bright channel cue. In *CRICV (with ECCV)*. Citeseer, 2010.

[21] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *TPAMI*, 25:918–923, 2003.

[22] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *CVIU*, 95(2):238–259, 2004.

[23] I. Sato, Y. Sato, and K. Ikeuchi. Illumination from shadows. *TPAMI*, 25(3):290–300, 2003.

[24] M. Shoaib, R. Dragon, and J. Ostermann. Shadow detection for moving humans using gradient-based background subtraction. In *ICASSP*, pages 773–776. IEEE, 2009.

[25] Y. Shor and D. Lischinski. The shadow meets the mask: Pyramid-based shadow removal. In *EuroGraphics*, volume 27, pages 577–586. Wiley Online Library, 2008.

[26] M. Szummer, P. Kohli, and D. Hoiem. Learning crfs using graph cuts. In *ECCV*, pages 582–595. Springer, 2008.

[27] E. Vazquez, R. Baldrich, J. van de Weijer, and M. Vanrell. Describing reflectances for color segmentation robust to shadows and textures. *TPAMI*, 33(5):917–930, 2011.

[28] J. Zhu, K. G. Samuel, S. Z. Masood, and M. F. Tappen. Learning to recognize shadows in monochromatic natural images. In *CVPR*, pages 223–230, 2010.