Fast Shadow Detection from a Single Image Using a Patched Convolutional Neural Network

Sepideh Hosseinzadeh, Moein Shakeri, Hong Zhang

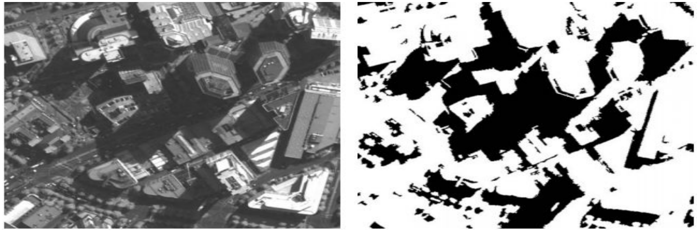
Oct, 2018

UNIVERSITY OF ALBERTA

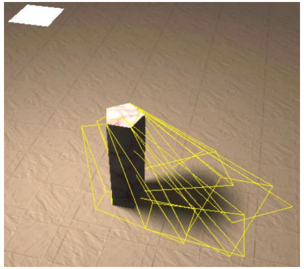
Motivations

Robotic Applications i.e. increasing awareness for auto-driving cars

Object Tracking



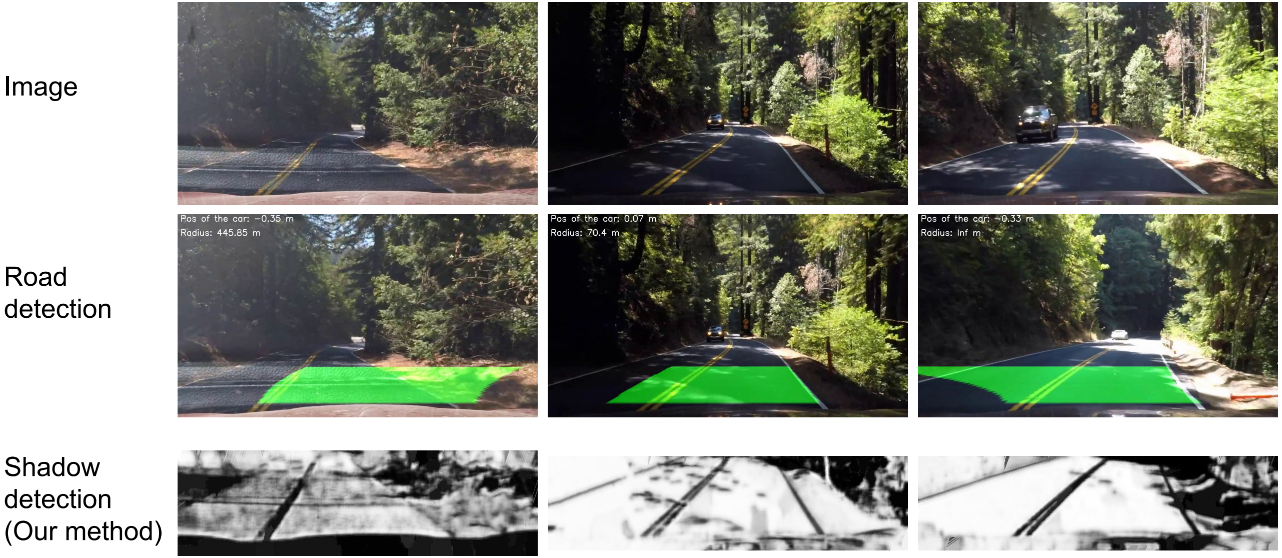
Aerial Imaging



Object/Shape recognition

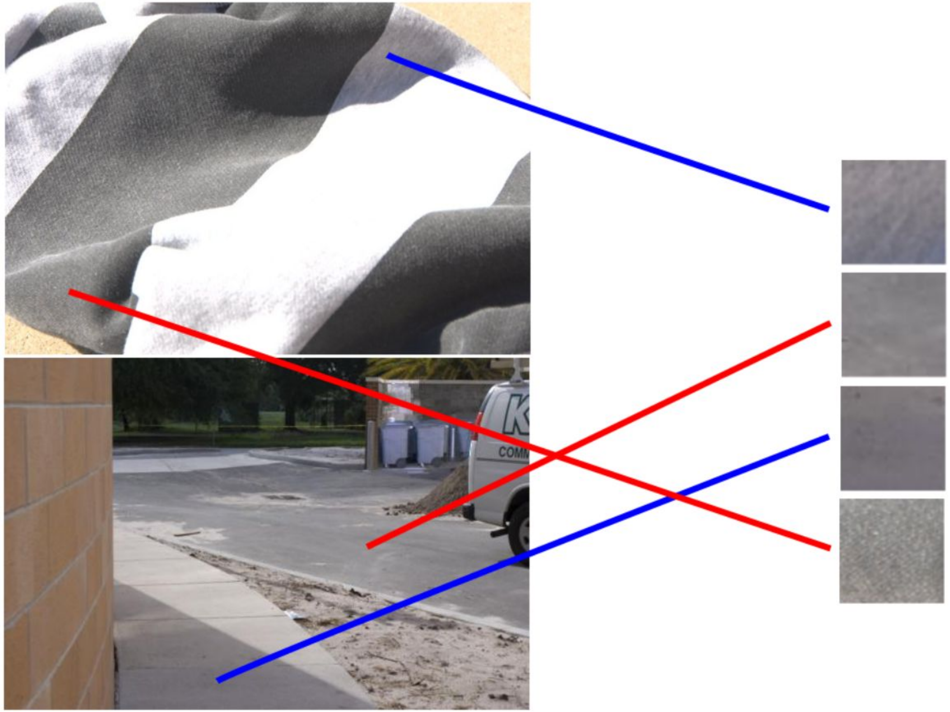
Motivations

- Shadows cause complications for **road detection** in **autonomus cars**.



Challenge

- Complex interactions of geometry, albedo, and illumination.
- Locally, we cannot tell if a surface is dark due to shading or albedo.



* Guo, Ruiqi, Qieyun Dai, and Derek Hoiem. "Paired regions for shadow detection and removal." TPAMI, 2013.

Related Works

1. Physical models of illumination and colour.

   * Finlayson et al. [1] used a 1-d shadow free image, to compare shadow edges.

   * *Weakness: need of planckian lighting, and high-quality input images.*

2. Statistical learning based approaches.

   * To learn shadow model parameters at each pixel from a video sequence.

   * *Weakness: need of video sequence for learning.*

3. Data-driven learning approaches for single-images.

   * Classifier trained on local hand-crafted features.

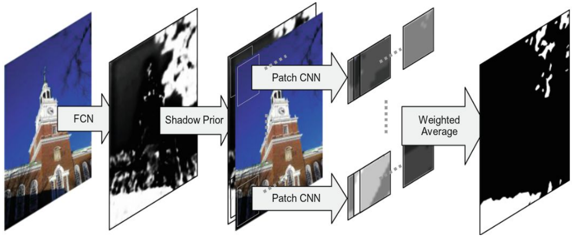
   * Lalonde et al. [2] detected cast shadow edges on the ground.

   * *Weakness: per-pixel outputs are inherently noisy with poor contour continuity.*

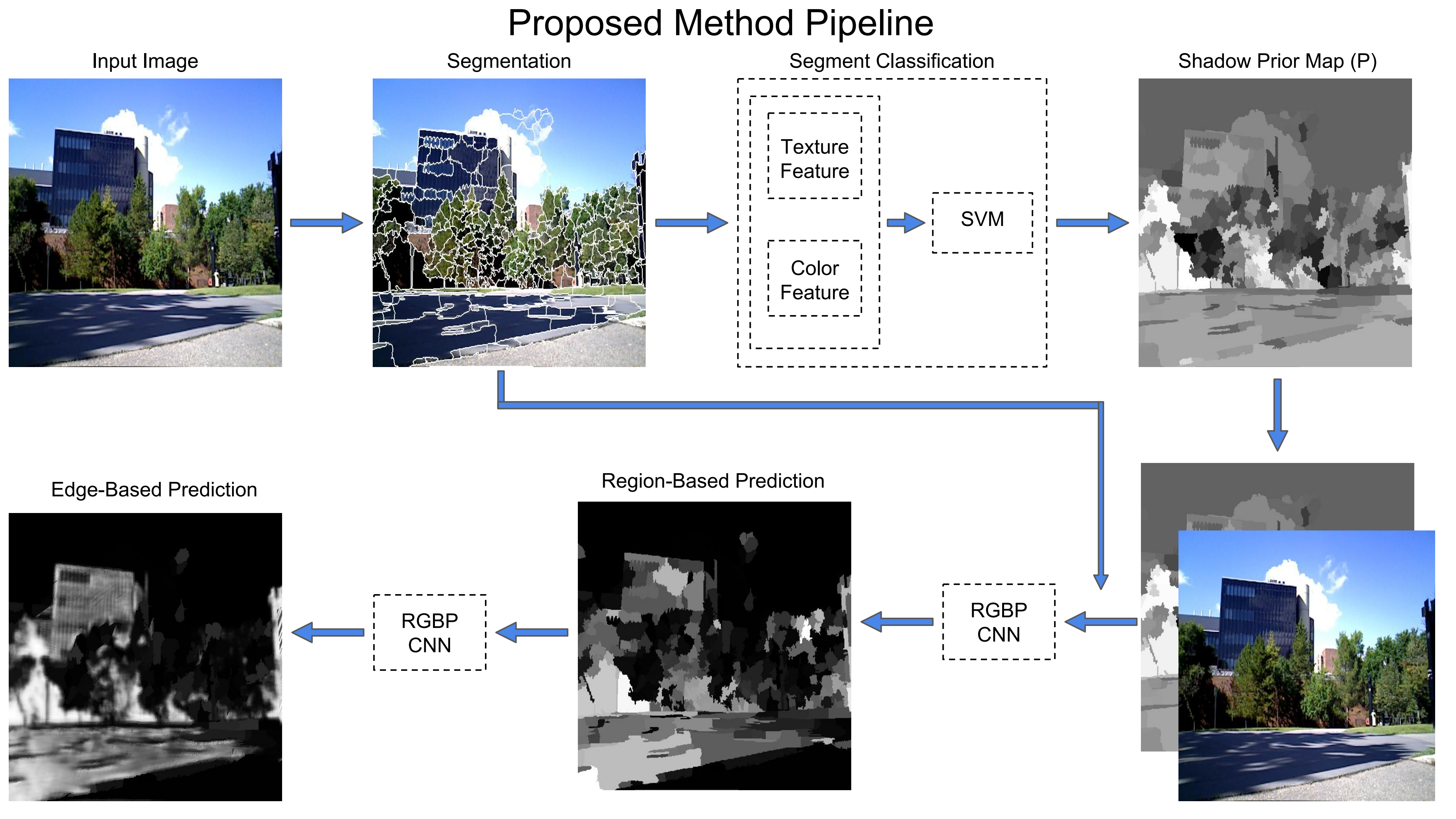
4. Deep Learning frameworks.

   4.1. Local prediction for each pixel, then they combined by optimization.

   4.2. Using prior knowledge beside image, Vicente, et al [3].



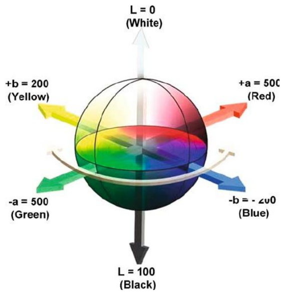
[1] G. Finlayson, et al. Removing shadows from images. ECCV. 2002.
[2] J.-F. Lalonde, et al. Detecting ground shadows in outdoor consumer photographs. ECCV, 2010.
[3] Vicente, Tomás F. Yago, et al. "Large-scale training of shadow detectors with noisily-annotated shadow examples." ECCV, 2016.


Proposed Method Pipeline
Input Image
Segmentation
Segment Classification
Texture Feature
Color Feature
SVM
Shadow Prior Map (P)
Edge-Based Prediction
Region-Based Prediction
RGBP CNN
RGBP CNN

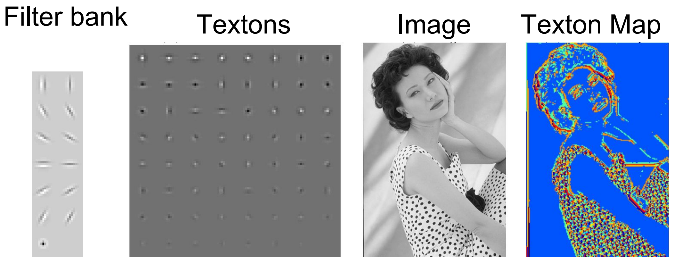
Shadow Prior Map

Steps of production:

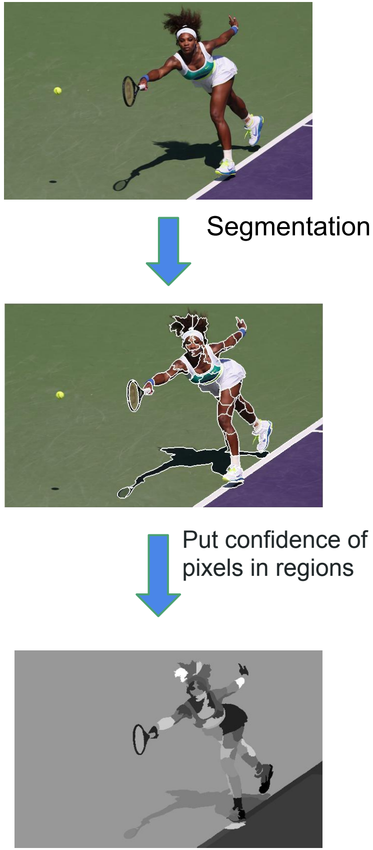
1. Image Segmentation by Mean-Shift.
2. Represent **color** with a histogram in L*a*b space, with 21 bins per channel.



3. Represent **texture** with the texton histogram.
    a. Vector of filter responses to every pixel.
    b. Vectors are clustered using k-means.
    c. Each pixel is assigned to nearest cluster center, or *texton*.



4. Train our classifier using an SVM.
5. Define confidence of each region = (output of SVM) × (pixel area of the region).



Why not FCN Prior map/Optimization?

1. FCN prop map prior knowledge is a **poor map**. It only considers **color of pixels**.

   a. It is so **slow** in training and testing.

2. Optimization methods are expensive.

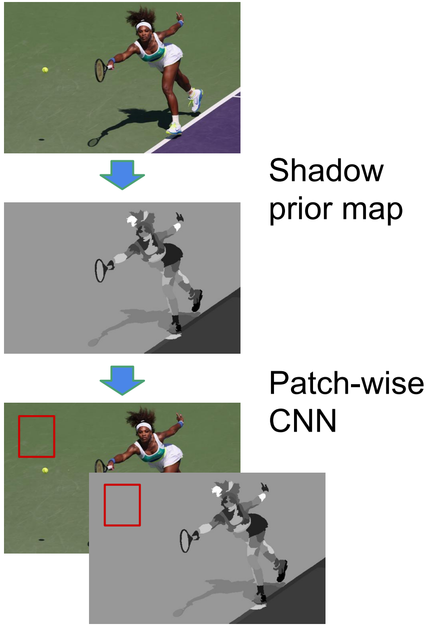
   a. Optimizing many parameters.

[1] Vicente, Tomás F. Yago, et al. "Large-scale training of shadow detectors with noisily-annotated shadow examples." ECCV, 2016.

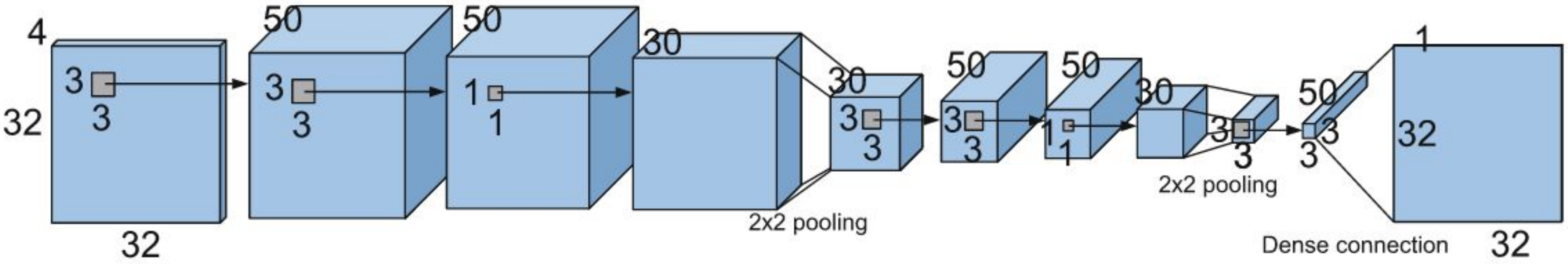
Training the RGBP CNN

Steps:

1. Predict a shadow prior map.
2. Map is attached to the RGB image.
3. Training a CNN on RGBP patches.



Network Structure



We used a basic CNN structure which has:

- 6 Convolutional layers,
- 2 Pooling layers,
- 1 Fully connected layer.
- The size of the input (patch size) is 32 × 32.

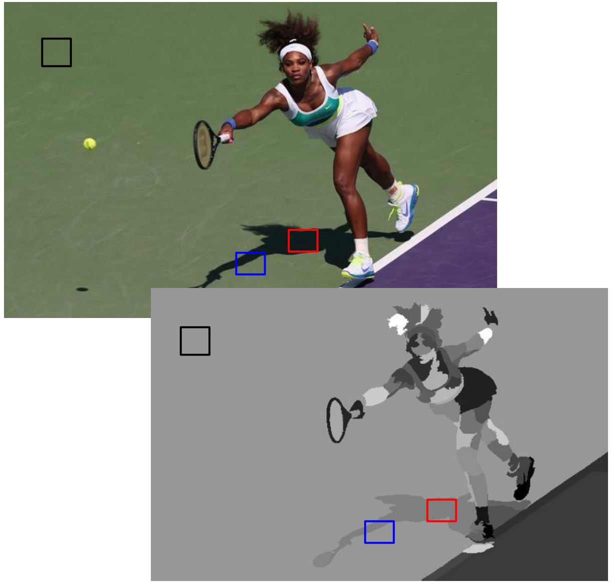
Selection of Patches

Guarantee that CNN learn various types of material:

1. On random non-shadow location,
   **patches of various textures and colors.*

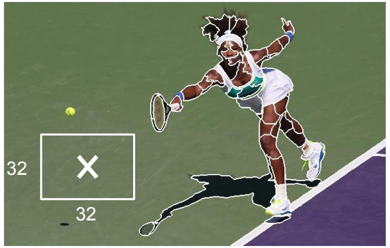
2. On Canny-edges between shadow and non-shadow regions,
   **hard-to-classify boundaries, and learn edge pixels.*

3. Shadow locations,
   **guarantee a minimum number of positive instances.*



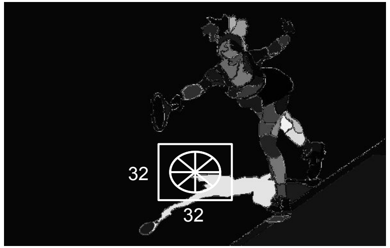
Proposed Method Algorithm

1. Computing shadow prior map.
2. Region-based prediction:

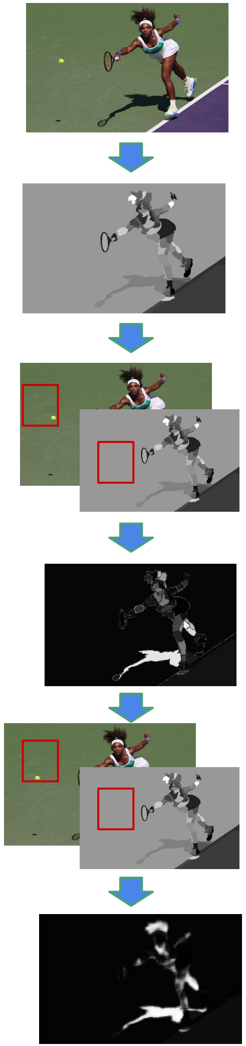


CNN → Mean of all patch pixel

3. Edge-based prediction: *(relate segments)*
   - Detect edges.
   - Refine info on the edge points and their neighbors.



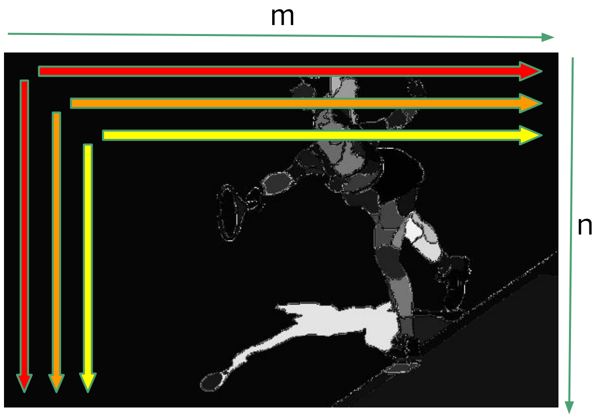
CNN → Mean of edge pixel and 8 neighbors



Time Complexity Analysis

- Reasons of having low time complexity:
  a. Batch processing, instead of single pixel processing.
     ** 1 pixel/segment*
  b. Processing only edge pixels, instead of all.
  c. Parallel computing by CNN.
  d. Time of processing = min(n,m) x O(CNN)

Batch-processing by using segmentation



Region-based detection

Methods of Evaluation

- Accuracy of shadow pixels = TP / P
- Accuracy of non-shadow pixels = TN / N
- Accuracy of pixels = (TP+TN) / (P+N)

Datasets

1. UCF Shadow Dataset [1]: 355 images. 120 train / 235 test.
2. UIUC Shadow Dataset [2]: 108 images. 76 train / 32 test.
3. SBU Shadow Dataset [3]: 4727 images. 4089 tarin / 638 test.
4. Combined Dataset of [1,2,3]: 5078 images. 3808 train / 1270 test (random 25% of images).

[1] J. Zhu, et al. Learning to recognize shadows in monochromatic natural images. CVPR, 2010.
[2] R. Guo, et al. Paired regions for shadow detection and removal. TPAMI, 2012
[3] Vicente, Tomás F. Yago, et al. "Large-scale training of shadow detectors with noisily-annotated shadow examples." ECCV, 2016.

Methods of Comparison

1. Unary-Pairwise:
   Guo et al. *"Paired regions for shadow detection and removal."* TPAMI, 2013.

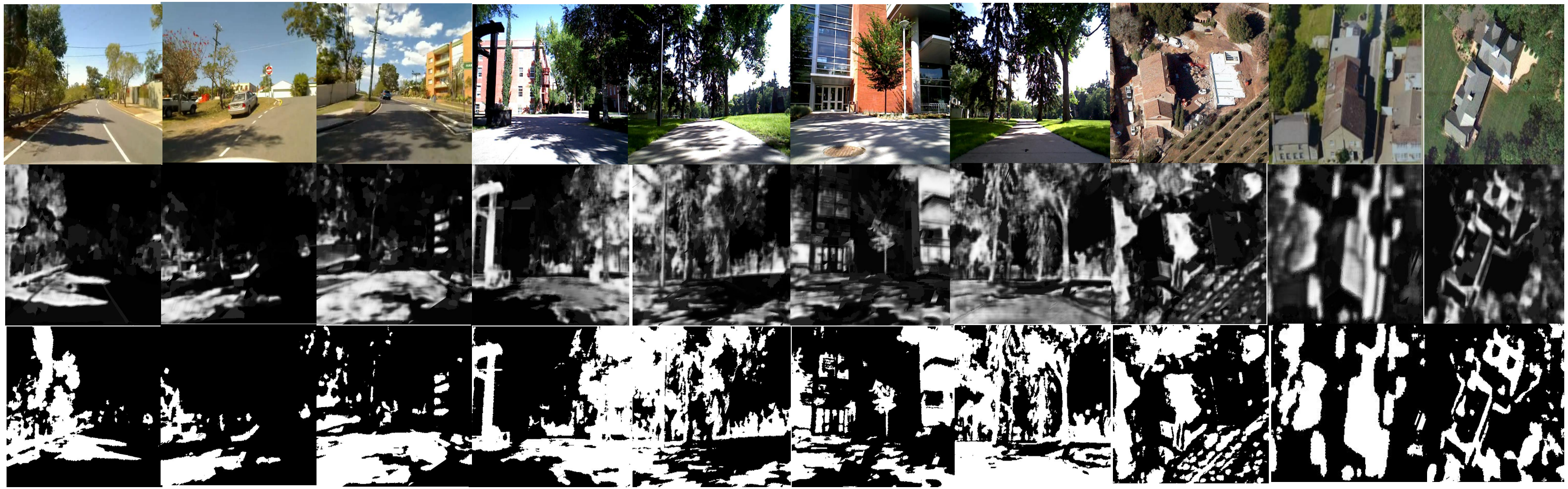
2. Stacked-CNN:
   Vicente et al. *"Large-scale training of shadow detectors with noisily-annotated shadow examples."* ECCV, 2016.

Qualitative Results

Image

Our Method

Our Method Binary



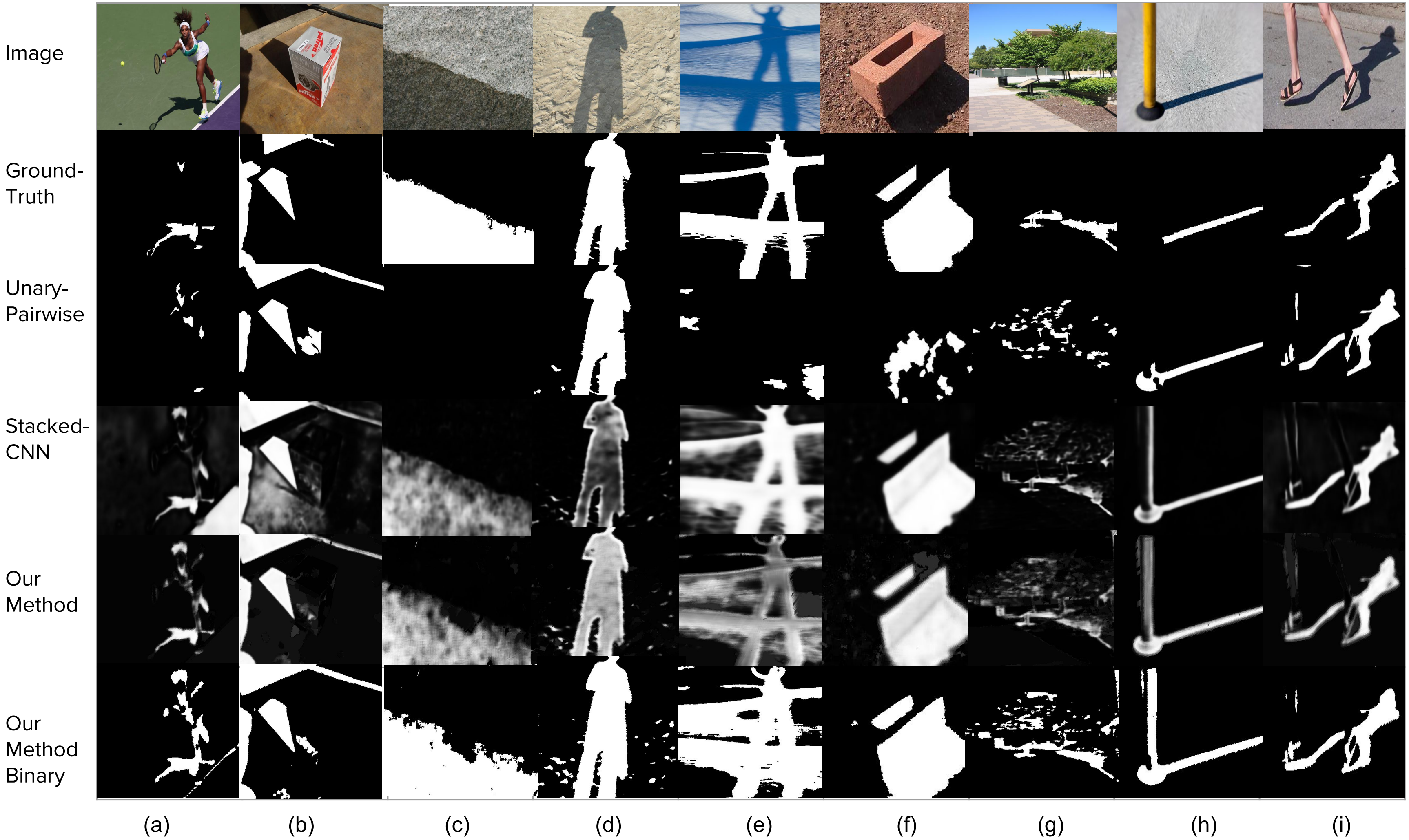


Image
Ground-Truth
Unary-Pairwise
Stacked-CNN
Our Method
Our Method Binary
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

Accuracy/Time Comparison

Method	Accuracy / std	Shadow acc	Non-shadow acc
Stacked-CNN	0.9044 / 0.12	**0.8614 / 0.18**	0.9140 / 0.13
Unary-Pairwise	0.8835 / 0.13	0.6374 / 0.32	**0.9366** / 0.11
Our method	**0.9103 / 0.11**	0.8527 / 0.20	0.9248 **/ 0.11**

Method	Test	Train
Stacked-CNN	27.77 h (78.75 sec/image)	9.08+(Training of FCN) h
Unary-Pairwise	20.77 h (58.87 sec/image)	-
Our method	**0.55 h** (1.55 sec/image)	**3.05 h**

Combined Dataset

*On single GeForce GTX 1080 Ti

Accuracy/Time Comparison

Method	**Accuracy / std**	**Shadow acc**	**Non-shadow acc**
Stacked-CNN	**0.8850 / 0.1362**	0.8609 / 0.2361	0.9059 / 0.1504
Unary-Pairwise	0.8639 / 0.1431	0.5636 / 0.3543	**0.9357 / 0.1253**
Our method	0.8664 / 0.1417	**0.8987 / 0.2074**	0.8773 / 0.1577

Method	**Test**	**Train**
Stacked-CNN	25.08 h (141.56 sec/image)	9.4 h+(Training of FCN) h
Unary-Pairwise	9.13 h (51.56 sec/image)	-
Ours	**0.33 h** (1.87 sec/image)	**3.1 h**

SBU shadow dataset

Conclusions

- Proposed a real-time and novel method for shadow detection from single images.
- Performs at the local patch level, and it can make use of shadow semantic information beside RGB images.
- Accuracy is close to state-of-the-art statistical and deep methods.

Thanks!