Seeing What Is Not There: Learning Context to Determine Where Objects Are Missing

Jin Sun David W. Jacobs
Department of Computer Science
University of Maryland
{jinsun,djacobs}@cs.umd.edu

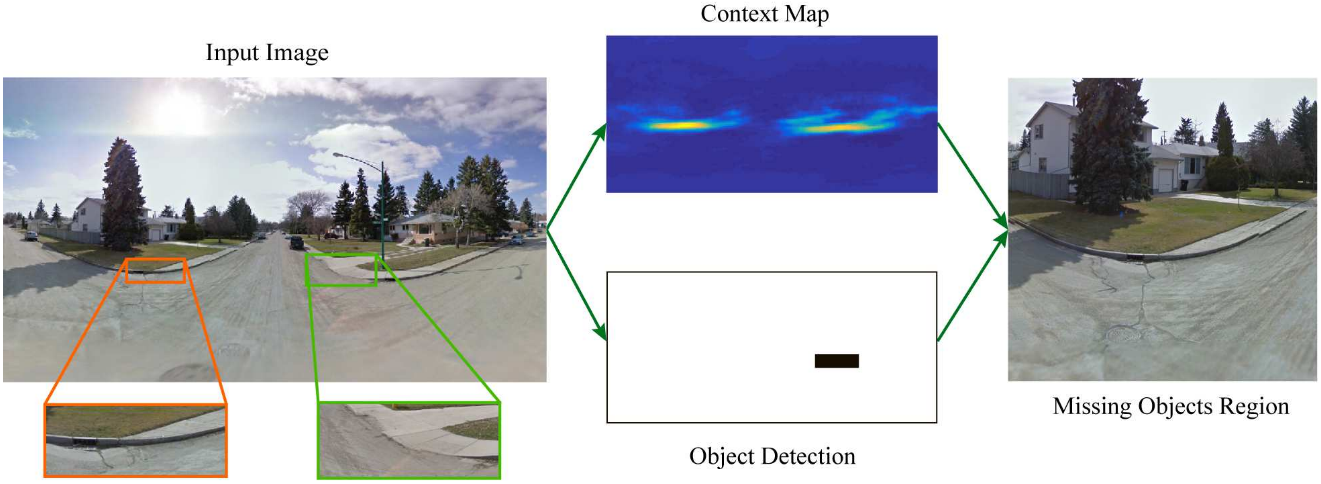

Figure 1: When curb ramps (green rectangle) are missing from a segment of sidewalks in an intersection (orange rectangle), people with mobility impairments are unable to cross the street. We propose an approach to determine where objects are missing by learning a context model so that it can be combined with object detection results.

Abstract

Most of computer vision focuses on what is in an image. We propose to train a standalone object-centric context representation to perform the opposite task: seeing what is not there. Given an image, our context model can predict where objects should exist, even when no object instances are present. Combined with object detection results, we can perform a novel vision task: finding where objects are missing in an image. Our model is based on a convolutional neural network structure. With a specially designed training strategy, the model learns to ignore objects and focus on context only. It is fully convolutional thus highly efficient. Experiments show the effectiveness of the proposed approach in one important accessibility task: finding city street regions where curb ramps are missing, which could help millions of people with mobility disabilities.

1. Introduction

Most fundamental computer vision tasks, e.g., image classification and object detection, focus on seeing what is there: for example, is there a curb ramp in this image, if yes, where is it? Using deep neural network models, computational approaches to such tasks are catching up to human performance in more and more benchmarks. However, humans can easily outperform algorithms in the task of inferring objects that are 'not there': for example, is there a curb ramp in this image, if no, where *could* it be?

We are interested in finding where objects are *missing* in an image: an object of interest is not there, even though the environment suggests it should be. From a computational perspective, an object can be defined as missing in an image region when: 1) an object detector finds nothing; 2) a predictor of the object's typical environment, i.e. context, indicates high probability of its existence. Given an image, we want to detect all such regions efficiently. We summarize the relationship between an object's detector and its context model in Table 1. While there are many existing works on utilizing context in object detection (Section 2), they mainly focus on improving performance on finding typical objects with contextual and object information entangled. In this work we propose to train a standalone object-centric context representation to find missing objects. By looking at the reverse conditions, it can be adapted to find out of context objects too.

One practical motivation for finding missing objects comes from the street view curb ramp detection problem (Figure 1). The task is to label curb ramps in a city's intersections so that people with mobility impairments can plan their routes with confidence. Although existing work [9] shows good performance in detecting constructed curb ramps, it cannot detect missing curb ramps regions. Knowing this information is highly valuable: users can assess the accessibility of an area; navigation algorithms can calculate better routes for pedestrians; governments can plan for future renovations accordingly. This is a very expensive and time consuming task for human labelers, which is partially the reason why such information is missing from public databases. Therefore, we are interested in developing an automatic algorithm that is effective and efficient. It can be used to scan a whole city to find regions where curb ramps are missing. In this scenario, the number of found true missing curb ramp regions (recall) is more important than precision because it is much more light-weight to ask humans to verify algorithm results than to label images from scratch. Moreover, even if the algorithm reports one true missing curb ramp region but mistakenly ignores three others in an image, it is still valuable as a preprocessing step: governments can prioritize intersection assessments in a city and allocate auditors more efficiently.

We believe the key to tackle this problem is to learn a model that focuses on context only and works efficiently just like an object detector: it scans each image and generates a probability heat map in which each pixel represents the probability that an object exists, even when no object is in sight. One big advantage of the context and object decomposition is that we don't need abnormal object labels (missing/out-of-context) for training. A standalone context model can be learned from typical objects and later used for finding abnormal objects. This greatly simplifies training: normal objects are abundant and much easier to collect and label than abnormal objects.

In this paper, we propose such a model based on convolutional neural networks and a novel training strategy to learn a standalone context representation of a target object. We start by introducing a base network in Section 3. It takes input images with explicit object masks and learns useful context from the remaining areas of the images. Because of the limitations discussed in Section 4, we then propose a fully convolutional version of the network that learns an implicit object mask such that it ignores objects in an image and focuses purely on context. It does not require object masks during test time. Section 5 describes the procedure for using the context model to find missing objects regions and Section 6 presents experimental results.

Object Score	Context Score	Image Region Remark
High	High	Typical objects
Low	High	Missing objects
High	Low	Out of context objects

Table 1: Relationship between object and context. Object score is obtained from an object detector, while context score is from its context model.

The contributions of this work are as follows. First, we propose a method to learn an object-centric context representation by learning from object instances with masks. Second, we propose a training strategy to force the network to ignore objects and learn an implicit mask. The model is fully convolutional so it also speeds up probability heat map generation significantly. Finally we present promising results on the missing curb ramps detection problem in street view images, and a preliminary result on finding out-of-context faces.

2. Related Work

Context in Object Recognition. A large body of evidence has shown that contextual information affects human visual search and recognition of objects [3, 12]. In computer vision, recently it also has become a well accepted idea that context helps in object recognition algorithms [5, 11, 13, 19]. Usually, context is represented by semantic labels around an object. [15] uses a Conditional Random Field to model contextual relations between objects' semantic labels to post-process object recognition results. [11] builds a deformable part model that incorporates context labels around an object as 'parts'. Because of the coupling between context and object information, these methods are unsuitable to detect missing object regions.

Torralba et al. proposed the Context Challenge [18] that consists in detecting an object using exclusively contextual information. They take the approach of learning the relation between global scene statistical features and object scale and position. Visual Memex [10] is a model that can either retrieve exemplar object instances or predict the semantic identity of a hidden region in an image. It uses hand-crafted features and models context as inter-category relations. Our approach can be seen as a general approach that attempts to address this challenge, without the need for designing hand-crafted features or using preset object classes.

Finding Missing Objects. Grabner et al. proposed to use the General Hough Transform to find objects that are missing in video frames during object tracking [7]. The idea is to estimate positions of a target object from surrounding objects with coupled motions.

Computer Vision with Masked Images. Recently Pathak et al. [14] proposed to learn a convolutional neural network context encoder for image inpainting. Both their work and ours train convolutional neural networks with masked images. But the purpose is very different as they try to learn a generative model to inpaint the mask while we learn a discriminative model to infer what is inside the mask. Also, our work uses an efficient fully convolutional structure.

Accessibility Tasks. With massive online resources such as the Google Street View (GSV) service, many computer algorithms are designed to help people with disabilities and improve their quality of life. CrossingGuard [8] is a system designed to help visually impaired pedestrians to navigate across intersections with help from Amazon Mechanical Turk. Tohme [9] is a semi-automated system that combines crowdsourcing and computer vision to collect existing curb ramp positions in city intersections using GSV images. It uses the Deformable Part Models [6] as a curb ramp detector and asks Mechanical Turkers to verify the results. They provide a street view curb ramp dataset with 1086 city intersection images, which we use in the experiment section.

3. Learning Context from Explicit Object Masks

In this section, we introduce a base version of the proposed context learning algorithm. If ‘context’ is defined to be everything that surrounds an object except the object itself, this model is learning context literally: every target object instance in training images is masked out. Here we assume an object’s visual extent is fully represented by its bounding box label.

This is a binary classification problem. Positive samples are collected so that each image sample has an object at its center, with a black mask (value equals zero after preprocessing) covering the object’s full extent. The bounding box width to the whole image width ratio is set to 1/4 for the purpose of including a larger contextual area. Negative samples are random crops with similar black masks at their centers. The position of a negative crop is chosen so that the masked region will not cover any groundtruth labeled objects with more than a Jaccard index [1] of 0.2.

If there are multiple object instances in an image, we mask out one object at a time for positive samples. This is because the existence of other object instances could be useful context: for example, curb ramps often appear in pairs.

To prevent our context model trivially learning the particular mask shape, we force negative samples to share a similar distribution of mask dimensions with positive samples. The sampling strategy is to interleave the positive sampling and negative sampling processes, and use the previous positive sample’s mask dimension in the next negative sample.

We train a convolutional neural network model Q. The network consists of four convolutional layers with pooling and dropout, and two fully connected layers. Its structure is summarized in Table 2. Cross entropy loss (Eq. 1) is used as the classification loss:

$$\mathcal{L}_c = -Q_y(I_m) + \log \sum_y e^{Q_y(I_m)}, \qquad (1)$$

where $y \in \{1, 2\}$ is the groundtruth label for a masked image I_m (1 for positive, 2 for negative), $Q(I_m)$ is a 2x1 vector representing the output from the network Q, while $Q_y(I_m)$ represents its y-th element.

Layer (type)	Shape	Param #
Convolution2D	(3, 3, 32)	896
Convolution2D	(3, 3, 32)	9248
MaxPooling2D	(2, 2)	0
Dropout	-	0
Convolution2D	(3, 3, 64)	18496
Convolution2D	(3, 3, 64)	36928
MaxPooling2D	(2, 2)	0
Dropout	-	0
FullyConnected	(53*53*64, 256)	46022912
Dropout	-	0
FullyConnected	(256, 2)	514
Total params: 46,088,994		

Table 2: Neural network structure summary for the base network. Convolution filter shapes are represented by (filter width, filter height, number of filters) tuples. The network expects to take an input image of size 224x224, with an explicit mask at the center.

Convolution2D	(53, 53, 256)	46022912
Dropout	-	0
Convolution2D	(1, 1, 2)	514

Table 3: Fully convolutional layers to substitute for the last three layers of the base network. This network can take arbitrary sized input, with no explicit mask needed.

During test time, a sliding window approach is used to generate a probability heat map for a new image so that each pixel has a context score of how likely it is to contain an object. At each position, a fixed size (224x224 in our implementation) image patch is cropped with the center region masked out to be fed into the base network. The mask size is determined empirically from the training set.

[1] Defined as the intersection-over-union ratio of two rectangles.

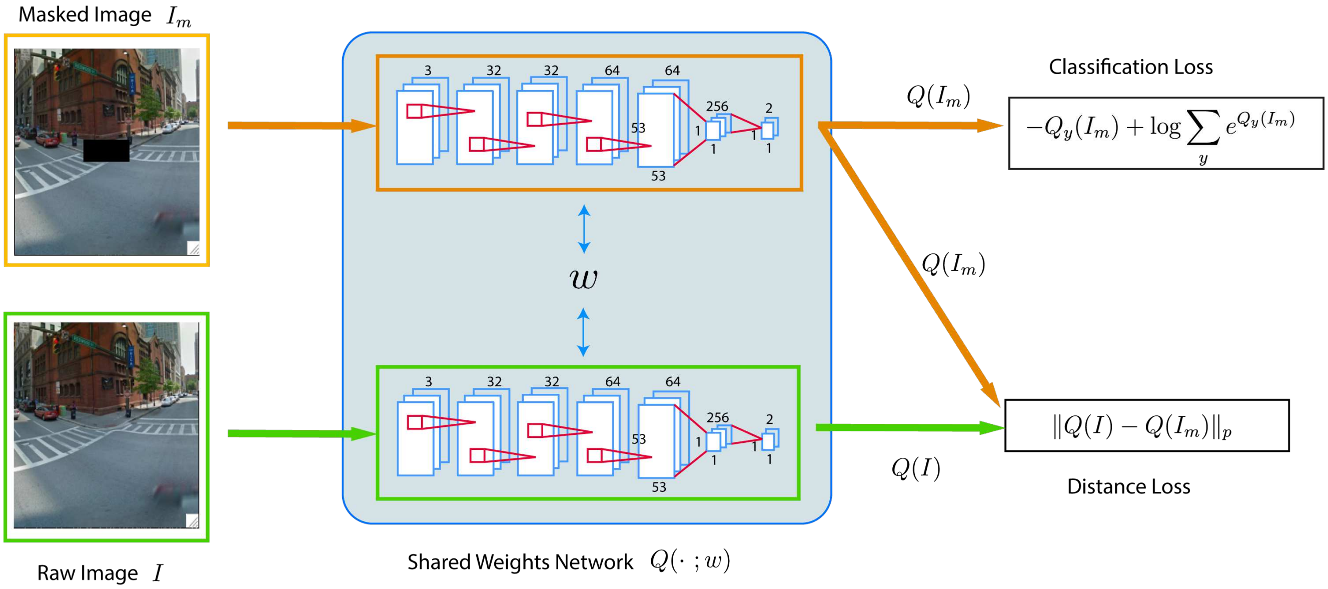

Figure 2: Training scheme of the **S**iamese trained **F**ully convolutional **C**ontext network (SFC). The intuition is to enforce the fully convolutional network Q to output similar results regardless of whether an image is masked or not. Additionally, the network should produce correct classification labels. The training is done in a Siamese network setting with shared weights w.

4. A Fully Convolutional Model that Learns Implicit Masks

There are several issues with a network trained with masked images. First, the network tends to learn artifacts. [14] reports that training with rectangular mask makes a network learn "low level image features that latch onto the boundary of the mask". They propose to use random mask shapes to prevent this issue. However, we cannot use random masks because our mask is defined over the visual extent of an object. Second, during testing time, the base network expects every input to have an explicit mask. This is highly inefficient when we evaluate the network at all positions and scales to generate a heat map. There are standard procedures to convert a convolutional neural network with fully connected layers into a fully convolutional one [17] so that the map generation is much more efficient for images of arbitrary sizes. However, in our case the situation is complicated. During training, the base network always sees input images with all zeros at the center, so the weights of neurons with receptive fields on this region can be arbitrary because no gradients are updated. If we apply the converted fully convolutional network to unmasked images, outputs from those neurons can affect the final map arbitrarily.

The question is then, can we train a network so that it is fully convolutional and learns context by ignoring the masked region 'by heart'?

The answer is yes and we now propose a training strategy to make a network learn an implicit object mask. The intuition is that we want the network to output similar results regardless of whether an input image is masked or not. By enforcing this objective, the network should learn to find visual features that are shared in both masked and raw images: i.e. from the unmasked regions.

Formally, we want to minimize a distance loss in addition to the classification loss used in the base network:

$$\mathcal{L}_d = ||Q(I_m) - Q(I)||_p, \tag{2}$$

where $Q(I_m)$ is the output vector from the network Q with masked image I_m as the input, $Q(I)$ is the output vector from Q with the unmasked raw image I as the input, and $\|\cdot\|_p$ represents the L_p-norm.

Effectively, we have two shared-weight networks that are fed with masked and raw image pairs (Figure 2). The network is a fully convolutional version of the base network (Table 3). One stream of the network computation takes a masked image as input and outputs $Q(I_m)$. In parallel, the other stream of network computation takes the unmasked raw image as input and outputs $Q(I)$. The classification loss $\mathcal{L}_c$ is calculated based on $Q(I_m)$ alone, while the distance loss $\mathcal{L}_d$ is calculated by $Q(I_m)$ and $Q(I)$. This structure is known as a Siamese Network [4] so we call it the Siamese trained Fully convolutional Context (SFC) network. Following [4], we choose the L_1 norm in distance loss $\mathcal{L}_d$. We expect the SFC network to learn an implicit object mask by assigning zero weights to neurons whose receptive field falls onto the center object mask region. During test time, unlike the base network, we don't have to manually set the mask size: the SFC network has encoded this information in convolutional filters' weights.

Finally, the overall training objective is defined as a weighted sum of the two losses:

$$\mathcal{L} = \lambda \mathcal{L}_d + \mathcal{L}_c, \tag{3}$$

where $\lambda = 0.5$ in our implementation.

The benefits of this training strategy are three fold:

1) Because the SFC learns to ignore object mask regions, we can directly apply it to new unmasked images with arbitrary sizes: it is now highly efficient to generate a dense probability map. Figure 3 shows a comparison between heat maps generated by the base network and the SFC network. A 1024x2048 pixels image costs about 5 minutes to generate a heat map with the base network while the SFC network takes less than 4 seconds to generate a map with higher spatial resolutions.

2) The SFC network is less prone to artifacts. It is possible for the base network to learn artifact features along the boundary of masks. Since such features are not present in unmasked images, the SFC network learns to ignore them.

3) During training, we can perform hard negative mining efficiently. Between each training epoch, we can apply the SFC network on all training images to generate heat maps and find high score false positive regions. Because of the efficiency of fully convolutional networks, this step can be easily included in training. Section 6.2 shows that hard negative mining indeed improves the network performance by a large margin.

5. Finding Missing Object Regions Pipeline

With a trained standalone context network (base network or SFC network), we summarize the procedure for finding missing object regions in a test image.

1) Generate a context heat map using the context network Q. This map shows where an object should appear.

2) Generate object detection results using any object detector. Convert detection boxes into a binary map by assigning 0 to the detected box region, 1 otherwise. This binary map shows where no objects are found.

3) Perform element-wise multiplication between the context heatmap and the binary map. The resulting map shows the regions where an object should occur according to its context but the detector finds nothing.

4) Crop the high scored regions (above a preset threshold) from the image according to the resulting map. These are the regions where objects are missing.

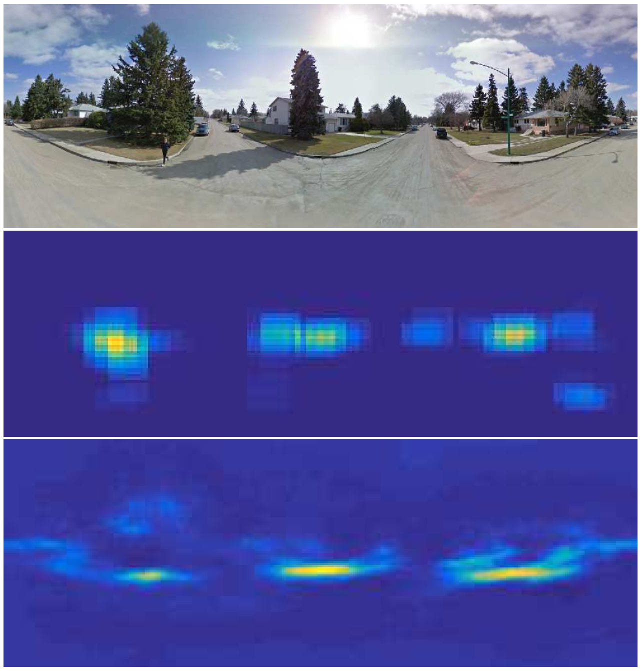

Figure 3: Top: an input street view panorama image. Middle: the heat map generated by the base network using a sliding window approach. Bottom: the dense heat map generated by the SFC network.

6. Experiments

In this section, we first examine the characteristics of the base network and the SFC network in Subsection 6.1. Then we evaluate their effectiveness. With the decomposition of context and object information, we study two unique tasks that can be efficiently performed using a standalone context model. Subsection 6.2 shows experimental results of finding missing curb ramp regions in street view images. Subsection 6.3 shows preliminary results of detecting out of context faces.

6.1. Characteristics of the Trained Model

As a validation study, we first check the sensitivity of the base and the SFC networks with regard to small changes in input images. All experiments are conducted on the curb ramp street view dataset. A desirable model has small response variations to the center region of an input image, where a mask was put during training. For evaluation, we change one pixel value at a time in a test image, by adding a small noise. The L_2 distance between a network's output before and after the disturbance is recorded for each pixel. In the end we obtain a map that shows which region in the image has large impact on the network's output. This can be seen as an estimate of the first order derivative of a network with respect to its input. Figure 4 shows the result with comparison between the base network and the SFC network. This result is summed over 20 different image samples.

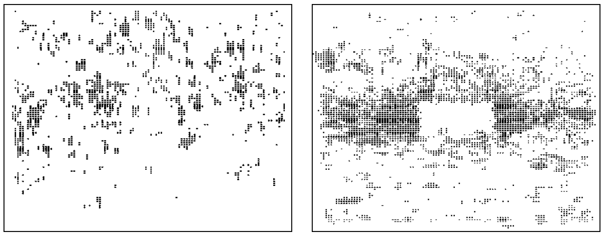

Figure 4: The sensitivity map of the base network (left) and the SFC network (right): a dark dot indicates a high sensitivity spot. Compared to the base network, the SFC map has a clear blank area at the center, which indicates that changes in this region have little effect on the network's output. The SFC network learns an implicit region mask.

From the result it is clear that the SFC network has small sensitivity at the center region of the input image. This is most likely due to the network learning to mute neurons whose receptive field falls at the center region of the input image. The blank region in the SFC's sensitivity map can be seen as a visualization of an approximation to the learned implicit region mask.

Next we check the distance loss $\mathcal{L}_d$ of the base network and the SFC network on test data. Using the same set of training hyper-parameters and setup (learning rate, training epochs) to train the two networks, the mean $\mathcal{L}_d$ loss is summarized in Table 4. It is clear that the SFC network is much more consistent in producing similar outputs regardless of object masks.

	SFC network	Base network
$\mathcal{L}_d$ loss	0.041	2.27

Table 4: Mean $\mathcal{L}_d$ loss of the two networks on the curb ramp dataset test set. Lower loss means smaller changes between a network's outputs from masked and unmasked images.

The SFC network works as intended: 1) it learns an implicit mask so it is less sensitive to any changes in the center region; 2) the useful features that it learns for the classification task are mainly from unmasked regions.

6.2. Finding Missing Curb Ramp Regions

Setup. We want to find missing curb ramps in the street view curb ramps dataset [9]. The dataset contains 1086 Google Street View panoramas which come from four cities in North America: Washington DC, Baltimore, Los Angeles and Saskatoon (Canada). Each panorama image has 1024x2048 pixels. It provides bounding box labels for existing curb ramps. On average there are four curb ramps per image. In addition, for our evaluation, an expert has labeled all missing curb ramp regions.

The dataset is split into half training and half testing. Each image is converted to YUV color space and normalized to be zero mean and one standard deviation in all channels. We use the curb ramp detector provided with the dataset, a Deformable Part Model, with default settings.

Training. For each epoch, 5000 samples are generated from training data, with half positives and half negatives. Figure 5 shows several examples. Each sample has 50% probability of being horizontally flipped for data augmentation purposes. Positive samples contain valid context around curb ramps. Negative samples are cropped randomly from areas not containing a curb ramp. To train the SFC network, each sample is prepared with two versions: raw and masked. We resize positive samples such that the object width is close to 55 pixels in a 224 pixels wide image. Each negative sample uses the same object mask and scale as the last positive sample to prevent the network overfitting to mask shapes.

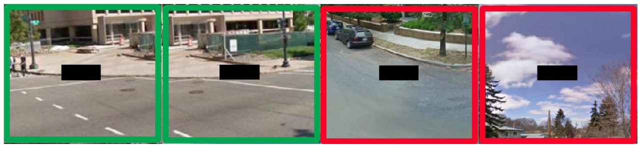

Figure 5: Training examples of curb ramps. Green rectangles represent positive samples, red rectangles represent negative samples.

We use the Keras/Tensorflow neural network software package [1]. The optimization algorithm uses Adadelta with default parameters. Since this is an adaptive learning rate method, there is no need to set a learning rate schedule during training. 20% of the training data is used as a validation set for an early stopping test. A base network and a SFC network are trained using the same hyper-parameters and training setup.

Results. Following the procedure described in Section 5, we run the two networks on test images to generate probability heat maps of where curb ramps should be in an image. For the base network, each heat map is generated in a sliding window scheme with a stride of 10 pixels, and various object mask widths of {50, 70, 100} pixels to generate multi-scale maps. The SFC network doesn't need an object mask size, so we resize the input panorama image with scales {0.5, 0.7, 1.0}. The numbers are chosen so that two networks see similar image pyramids. We use the DPM detector provided with the dataset to generate detection results. For each panorama, we generate a final map that combines the detection and context map and crop high scored regions (above a certain threshold) with size $d \times d$. According to preliminary empirical studies, we set the context threshold to 0.4 throughout the experiment.

We use human verification to evaluate the quality of the reported missing curb ramp regions. For that purpose, we develop a web based interface (Figure 6) that displays a gallery of found regions, ranked by their context scores. For each candidate region, a user provides feedback on whether it is truly a missing curb ramp region. We compare context maps generated by the base and the SFC networks with three baseline methods: random scores, spatial prior map, and a Faster RCNN [16] missing curb ramp detector.

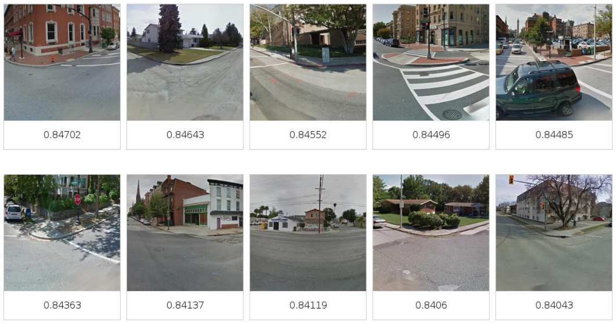

Figure 6: The web interface for verification. Each thumbnail shows one retrieved region, with its score displayed below. A user clicks on a thumbnail to verify it.

Random scores assigns uniformly random context scores from $[0, 1]$ to all positions in an image. This is a reference baseline showing the performance by chance.

A spatial prior map is built using the prior positions of curb ramps in street view panoramas. We use the prior map as a replacement for the context map for comparison. We collect the prior spatial distribution of all curb ramps from the training images. The collected distribution is smoothed with a 30x30 pixel Gaussian kernel with sigma=10. Figure 7 shows the spatial prior map used in our experiment. Because most panoramas are at street intersections, there is strong spatial structure consistency among the dataset. We expect this approach to be a reasonable baseline.

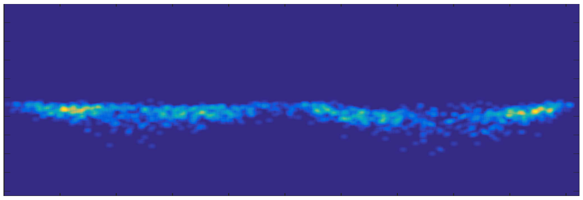

Figure 7: The spatial prior heat map generated from groundtruth locations of curb ramps in the training set. It shows that curb ramps are far from uniformly distributed.

With missing curb ramp region labels, we can treat this task as a standard object detection problem and directly train a Faster RCNN detector: the positive 'object' is a region labeled as missing curb ramps. Note that a Faster RCNN detector is capable of learning context because it's an end-to-end approach: potentially the detector can learn from the whole image to predict locations of missing curb ramp regions. We expect the Faster RCNN detector to be a strong baseline.

The verification of the missing curb ramp regions requires domain knowledge. One of the authors who has extensive experience with accessibility problems verified the results using our web interface. Figure 8 shows the comparison in recall of true missing curb ramp regions versus the number of visited regions (Recall@K). The retrieved region size is set to $d = 400$ pixels. 500 regions were retrieved from 543 test images.

The result shows that the SFC network with hard negative mining outperforms all other methods. We believe its superiority comes from the highly efficient fully convolutional structure that helps in training and generating high resolution context maps. Spatial prior map shows reasonable performance, which confirms the spatial bias of curb ramps locations in the dataset. Unlike the spatial prior map, the proposed methods can work well on other datasets that have no such bias. The Faster RCNN detector has significantly less recall compared with the SFC networks. With more missing curb ramp regions as training data, we expect the Faster RCNN detector to show improved performance; on the other hand, the SFC network does not even need missing curb ramp labels in training. The proposed methods learn useful context information from normal curb ramps, which are much easier to collect and label than missing curb ramp regions. Moreover, the SFC network is using detection results from a less advanced curb ramp detector (a DPM model shipped with the dataset): 77% of the false missing curb ramp retrievals are due to inaccurate curb ramp detections. Due to the page limit, we show more qualitative results of retrieved regions in the supplementary document.

Additionally, we investigate the effects of the retrieved region size d on the number of true missing curb ramp regions. Specifically, we vary the cropped region size from 400 pixels in width to 100 pixels. With smaller region size, it becomes crucial that the region is accurately localized with missing curb ramps at the center. Table 5 shows that the SFC network is not affected too much by the reduced field of view. This is because the regions it found are very well localized (See Figure 6). On the other hand, two baseline methods (random scores and prior maps) are performing poorly when the region size becomes small.

Discussion. Among 543 street view intersections in the test set, the SFC network is able to find 27% of the missing curb ramp regions by merely looking at 500 regions. This is an impressive result: 1) The whole process is very efficient (Table 6) such that it can be easily deployed to scan new city areas. For example, there are about 2,820 intersections

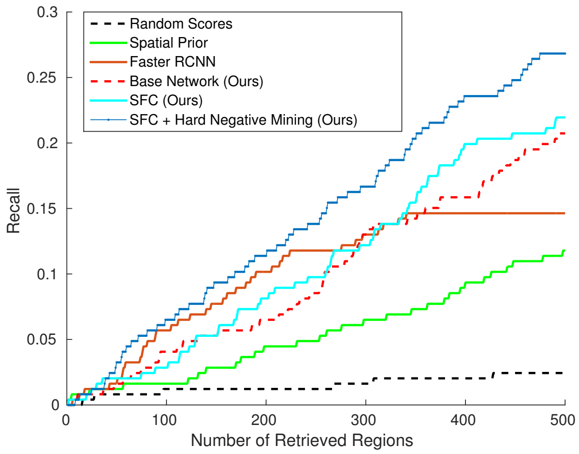

Figure 8: Recall of true missing curb ramp regions vs number of regions viewed (Recall@K). Our base and SFC networks outperform the two baseline methods (random scores and prior maps) by a large margin. The difference in recall between the Faster RCNN detector and the proposed method is substantial. The SFC network with hard negative mining has the best result among the proposed methods.

Region Width	400	200	100
SFC	35	33	27
Spatial Prior	13	8	4
Random Scores	4	2	0

Table 5: Effect of retrieved region size on the raw number of found missing curb ramps with 255 regions (the higher the better). As the region width shrinks, SFC performs very consistently while the two baseline methods (random scores and prior maps) suffer from poor localizations.

	Context Map (*)	Detection	Verification
Cost	4s/image	22s/image	20min/500 ims

Table 6: Time costs for different steps in finding missing curb ramps. The whole process is efficient as context and detection maps can be generated in parallel. *Using the SFC network.

in Manhattan, New York: it will take merely a few hours for our system to find missing curb ramps in a region with 1.6 million population; 2) Accessibility reports have shown that curb ramps condition (missing or not) shows high proximity consistency: if one intersection is missing curb ramps, it is highly likely that the nearby intersection has similar issues [2]. Our results can be used as an initial probe to quickly locate city areas that need special attention.

6.3. Finding Out of Context Faces

The pipeline in Section 5 for finding missing objects can be adapted to find out of context objects with just a few small modifications: change step 2 by assigning 1 to detected box regions and 0 for other regions; change step 4 to retrieve the lowest scored regions. Here we show a preliminary result of finding out of context faces to demonstrate both the generalization ability of the proposed method in different domains and possible future directions.

The task is to find out of context faces in the Wider face dataset [20]. Using a similar procedure as in finding missing objects and a state-of-the-art face detector [21], we retrieve the top 500 face regions that contain high face detector scores and low context scores from the validation set. For evaluation, we define an out of context face as a face without a visible body. Figure 9 shows qualitative results of the SFC network. We compare the SFC network results with random scoring. Out of 500 regions, the SFC network can find 27 out of context faces while random scoring found 14. While this result is preliminary, it suggests that the proposed method has the potential to be used in many other applications where finding out of context objects is important: for example, visual anomaly detection.

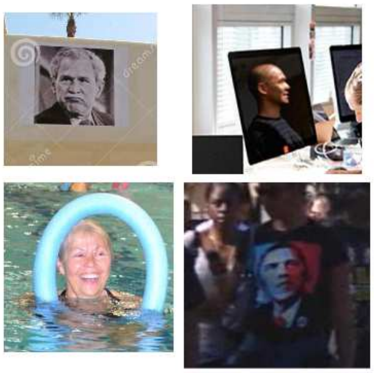

Figure 9: Retrieved out of context faces by a SFC network.

7. Conclusion

We present a approach to learn a standalone context representation to find missing objects in an image. Our model is based on a convolutional neural network structure and we propose ways to learn implicit masks so that the network ignores objects and focuses on context only. Experiments show that the proposed approach works effectively and efficiently on finding missing curb ramp regions.

Acknowledgments

This work was supported by NSF grants (IIS-1302338 and 1526234).

References

[1] Keras: Deep learning library for theano and tensorflow. https://keras.io/. 6

[2] Toward universal access: Americans with disabilities act sidewalk and curb ramp self-evaluation report. *City of Bellevue, WA*, September 2009. 8

[3] M. Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8):617–629, Aug. 2004. 2

[4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, June 2005. 4

[5] S. Divvala, D. Hoiem, J. Hays, A. Efros, and M. Hebert. An empirical study of context in object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009*, pages 1271–1278, 2009. 00155. 2

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sept. 2010. 3

[7] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1285–1292, 2010. 00062. 3

[8] R. Guy and K. Truong. CrossingGuard: Exploring Information Content in Navigation Aids for Visually Impaired Pedestrians. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 405–414, New York, NY, USA, 2012. ACM. 3

[9] K. Hara, J. Sun, R. Moore, D. Jacobs, and J. Froehlich. Tohme: Detecting Curb Ramps in Google Street View Using Crowdsourcing, Computer Vision, and Machine Learning. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, pages 189–204, New York, NY, USA, 2014. ACM. 2, 3, 6

[10] T. Malisiewicz and A. Efros. Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1222–1230. Curran Associates, Inc., 2009. 2

[11] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014. 2

[12] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 2

[13] W. Ouyang, X. Zeng, and X. Wang. Single-Pedestrian Detection Aided by Two-Pedestrian Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1875–1889, Sept. 2015. 2

[14] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 3, 4

[15] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007*, pages 1–8, 2007. 00365. 2

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 7

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2014. 4

[18] A. Torralba. Contextual priming for object detection. *IJCV*, 53:2003, 2003. 00516. 2

[19] P. Wang, L. Liu, C. Shen, Z. Huang, A. van den Hengel, and H. Tao Shen. What's wrong with that object? identifying images of unusual objects by modelling the detection score distribution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[20] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[21] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct. 2016. 8