Street-view Change Detection via Siamese Encoder-decoder Structured Convolutional Neural Networks

Xinwei Zhao[1], Haichang Li[2], Rui Wang[2], Changwen Zheng[2] and Song Shi[3]
[1]*Institute of Software Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China*
[2]*Science and Technology on Integrated Information System Laboratory, Institute of Software Chinese Academy of Sciences, Beijing, China*
[3]*Teleware Info & Tech (Fujian) Co.,LTD, Fujian Province, China*

Keywords: Change Detection, Semantic Segmentation, Siamese Network, Deep Learning.

Abstract: In this paper, we propose a siamese encoder-decoder structured network for street scene change detection. The encoder-decoder structures have been successfully applied for semantic segmentation. Our work is inspired by the similarity between change detection and semantic segmentation, and the success of siamese network in comparing image patches. Our method is able to precisely detect changes of street scene at the presence of irrelevant visual differences caused by different shooting conditions and weather. Moreover, the encoder and decoder parts are decoupled. Various combinations of different encoders and decoders are evaluated in this paper. Experiments on two street scene datasets, TSUNAMI and GSV, demonstrate that our method outperforms previous ones by a large margin.

1 INTRODUCTION

Change detection, i.e., finding meaningful changes from registered image pairs of the same region but captured at different time, is an important task in computer vision. Specifically, given the registered image pairs, we need to label each pixel as positive if it has changed at semantic level or negative otherwise, and produce a change mask at last, as Fig. 1 shows. Change detection has been widely applied in several areas including sandy land monitoring, offshore oil spill detection, and urban planning, etc.

Change detection is quite challenging as a lot of factors introduce irrelevant visual differences to the image pairs, such as the differences in shooting equipment, shooting conditions and weather. An example of street-view image pairs is shown in Fig. 1. In Fig. 1, the appearance of the buildings is quite different as they are captured in different weather. This demonstrates that objects in such image pairs are likely to show large variability even if they are unchanged.

Change detection has been studied for decades (Singh, 1989). Most methods are pixel-based such as image differencing and change vector analysis. In most cases, these methods are unable to exclude the aforementioned irrelevant visual differences, and cannot precisely detect the changes as desired. However, convolutional networks are robust to handle these problems. In this paper, we will explore street-view change detection using deep learning methods.

The goal of this paper is to propose a method to detect semantic changes precisely from registered street-view image pairs at the presence of the irrelevant changes. Inspired by the development of semantic segmentation (Long et al., 2015; Chen et al., 2017; Chen et al., 2018), we build a siamese encoder-decoder structured convolutional network (SEDS-CNN) to handle the change detection problem. The flowchart of the SEDS-CNN is shown in Fig. 2.

Specifically, the two encoders of the siamese convolutional networks have an identical structure and share the same weights, and the two decoders have the same characteristics. They are designed to extract the semantic information of image pairs. The last part of SEDS-CNN, the differentiator, takes the absolute difference of two feature maps generated by the decoders and produces 2-channel feature maps to denote the probability of changes and non-changes.

Moreover, we explore different combinations of encoders and decoders to construct the SEDS-CNN. Experiments on two publicly available datasets TSUNAMI and GSV (Sakurada and Okatani, 2015) show that the proposed SEDS-CNN model outperforms the

Zhao, X., Li, H., Wang, R., Zheng, C. and Shi, S.
Street-view Change Detection via Siamese Encoder-decoder Structured Convolutional Neural Networks.
DOI: 10.5220/0007407905250532
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 525-532
ISBN: 978-989-758-354-4

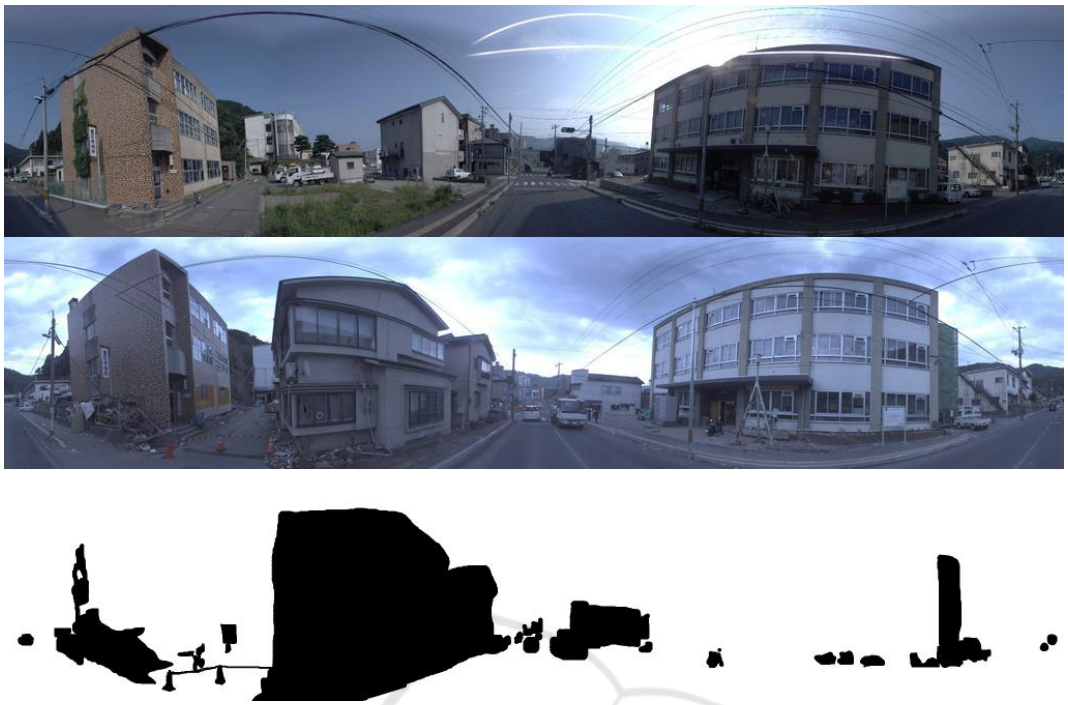


Figure 1: One sample from TSUNAMI dataset (Sakurada and Okatani, 2015). The top and middle images are the same place captured at different time, and the bottom image is the corresponding change mask. Black blocks are changed regions while white ones are not changed. Note that although the two images on the top show large variability because of the sunlight, most regions including the buildings on the left and right side are not regarded as changes.

existing approaches by a large margin.

The main contributions of this paper are listed as follows:

1. A siamese network, called SEDS-CNN, is proposed for street-view change detection. This network is an end-to-end framework, which can extract semantic information and predict changes at semantic level.
2. Experiments on two typical street-view datasets demonstrate the effectiveness of the proposed method, showing our model's robustness to irrelevant visual differences. Our method outperforms the previous ones by a large margin.

The remainder of our paper is organized as follows: Section 2 provides an overview of related work on change detection and semantic segmentation. Section 3 presents the structure of our networks. Section 4 details the experiment configuration and analyzes the results. Section 5 concludes our work.

2 RELATED WORK

2.1 Change Detection

Change Detection has been studied for several decades (Singh, 1989; Radke et al., 2005; Tewkesbury et al., 2015) and witnessed great success in many areas especially remote sensing. Among these conventional methods, the pixel and post-classification methods still remain popular even they have been proposed for nearly 30 years. While these algorithms produce good results on remote sensing images, their robustness is far from enough to overcome the irrelevant visual differences in optical image pairs, whose final appearances are much more easily affected by uncontrollable factors such as changing weather and seasons, or small camera displacement. A number of other methods focus on probability graph model including Markov Random Field (Bruzzone and Prieto, 2000), Conditional Random Field (Li et al., 2018), and Restricted Boltzmann machine (Gong et al., 2016).

In recent years, deep convolutional neural networks have shown striking power in computer vision. Since then, more sophisticated CNNs are proposed

to tackle several classic vision problems. SSD (Liu et al., 2016), YOLO (Redmon and Farhadi, 2017) and FPN (Lin et al., 2017) are proposed to solve object detection problems. FCN (Long et al., 2015), Deeplab(Chen et al., 2018) and many other fully convolutional based methods are designed to deal with semantic segmentation. All these networks perform far better than those methods that don't use deep learning. However, few deep learning models have been proposed to deal with change detection, at least for supervised methods.

(Zagoruyko and Komodakis, 2015) proposed a siamese CNN architecture to compare image patches to identify if objects in two images captured from different angles are the same one. Although it is not impossible to compare the entire image pairs in a patch-by-patch way, it's nearly unpractical both in terms of complexity and performance. (Sakurada and Okatani, 2015) used a VGG16 network fully pretrained for large-scale object recognition task to help extract features, and put them together with some manually designed features to detect changes in street scene image pairs. (Alcantarilla et al., 2018) proposed a deconvolutional neural network to perform change detection in street-view images. They superimposed one image on top of the other to obtain a 6-channel image, and then feed it to a simple deconvotlutional network, which comprises 4 convolutional and 4 deconvolutional layers. Although deep learning are used in these models, successful techniques in semantic segmentation such as dilated convolution and multi-scale pyramid pooling(Zhao et al., 2017; Chen et al., 2018) are not applied. Besides, they didn't merge low-level feature maps either, which have rich boundary information.

2.2 Semantic Segmentation

Deep convolutional networks have been successfully applied both in recognition and semantic segmentation. (Long et al., 2015) was the first to propose the fully convolutional neural network (FCN) trained end-to-end to solve the dense pixel-wise prediction tasks. However, the loss of spatial information caused by pooling layers is the major reason to restrict its performance. In order to tackle this problem, several techniques are proposed to preserve spatial information as the network goes deeper. (Yu and Koltun, 2015) proposed dilated convolution to replace pooling and convolutional layers at the latter part of convolutional networks, and it indeed expands the receptive fields and preserves the resolution of feature maps at the same time, without increasing the number of parameters. (Ronneberger et al., 2015) suggests concatenating low-level features to high-level ones to compensate for the loss of spatial information. (Chen et al., 2017) proposed to use global pooling and ASPP (Atrous Spatial Pooling Pyramid) to capture multi-scale information. Based on that, (Chen et al., 2018) merged low-level feature maps to ASPP, and obtained the state-of-art semantic segmentation model evaluated on PASCAL VOC dataset.

Most proposed networks for semantic segmentation can be explained from an encoder-decoder perspective, in which encoders are used to extract spatial and semantic information while decoders are used to gather them to give each pixel a semantic label. This is quite similar to change detection, where each pixel is labeled changed or unchanged. In this case, we can regard changes or non-changes as a kind of semantic label, and this is the motivation behind our approach: use encoder-decoder structures as the backbone for our model.

3 PROPOSED MODEL

In this section, we present the proposed SEDS-CNN model. The overall flowchart is shown in Fig. 2. From Fig. 2, we can find that the network has three parts: encoder, decoder, and differentiator. The structures of the first two parts are identical and share the same parameters. This is because (1) the two images from each pair are unordered and we can not specify which image precedes the other one, and (2) two images of each pair should be projected to the same semantic feature space to be compared. The extracted semantic features produced by decoders are then employed by the differentiator and the following components for change detection. We will describe the above three parts in detail in the following paragraphs.

3.1 Encoder

The appearance for the same object could be variable in different images, even if they are unchanged at semantic level. Inspired by this point, we intend to project the original RGB image into the semantic feature space, which is favorable for change detection. The above idea can be achieved by the encoding step.

The encoder part of SEDS-CNN model consists of multiple convolutional layers, dilated convolutional layers and max pooling layers. Their functions are merely to generate semantic features from the images. As the input images go through these layers, the extracted features become more and more abstract, which have more semantic information.

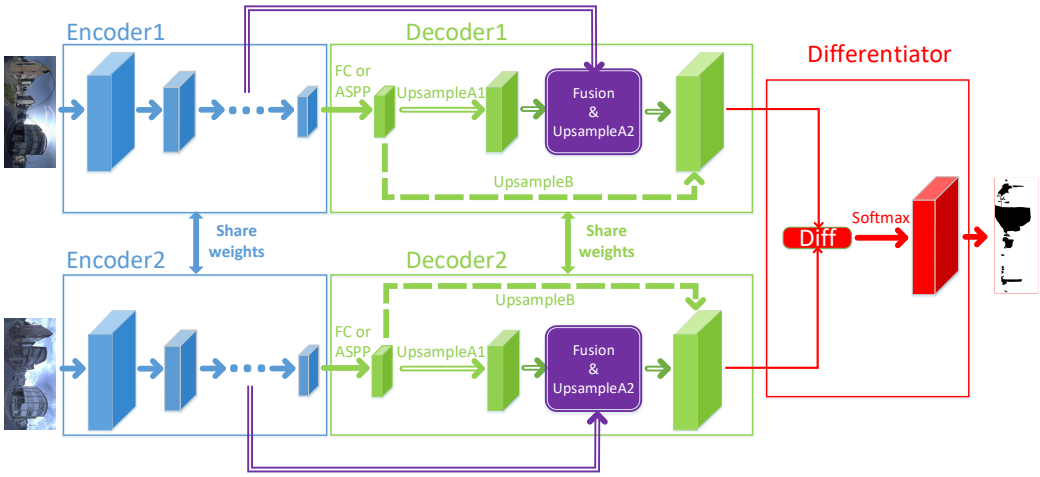


Figure 2: Overview of the encoder-decoder structure, which is composed of three parts. (1) Encoder: takes an image as input and produces multiple feature maps of different resolution. (2) Decoder: gathers feature maps from the encoder and recovers spatial information. There are two subbranches in the decoder: UpsampleA1 followed by Fusion & UpsampleA2 and UpsampleB. The green and purple double-line arrows traverse one subbranch and the dashed green arrows traverse the other. One of the two subbranches should be chosen and must be consistent in the two decoder modules. (3) Differentiator: take the absolute difference of the two recovered feature maps from last part, feed it to a softmax layer, and produce a change mask.

Unlike CNNs used in object recognition (Simonyan and Zisserman, 2014; He et al., 2016) which discard all their spatial information and produce high-level semantic class labels, CNNs for change detection need to predict semantic changes for each pixel. Thus, the spatial information has to be maintained, and it's same as in semantic segmentation. Therefore, the decoder part is necessary for change detection model.

3.2 Decoder

While encoders are used to produce high-level and low-level features, decoders gather them to recover the spatial information and produce the change mask. In our work, we employ FC (Fully Convolutional layer) from Fully Convolutional Network(Long et al., 2015) and ASPP (Atrous Spatial Pyramid Pooling) from Deeplabv3 (Chen et al., 2017; Chen et al., 2018) as the main components of our decoders.

FC merely reduces the channel of the input feature maps while ASPP uses 6-dilated, 12-dilated and 18-dilated convolutional kernels along with a global pooling layer to extract multi-scale features. As both of the two modules take the last layers from the encoders as input, detailed spatial information such as boundaries is lost. Therefore, (Long et al., 2015) upsamples the feature maps at stride 32, and adds feature maps at stride 8 from the 3rd pooling layers and feature maps at stride 16 to them to compensate for lost spatial information. Similarly, (Chen et al., 2018) concatenate stride 4 feature maps to upsampled stride 32 feature maps produced by ASPP module.

Table 1: Detailed settings for differnet decoders. *A1 refers to UpsampleA1.

Decoder	target size of A1*	Fusion
FC	stride 16	addition
ASPP	stride 4	concatenating

Following their implementations, we deploy FC and ASPP as exactly as they do. As Fig. 2 shows, within the decoder module, two subbranches are provided. One subbranch fuses feature maps from lower levels and the other doesn't. Any one of these two subbranches can be chosen. Fusion model fuses low-level features to compensate for detailed boundary spatial information.

Different settings are detailed in Table 1 when FC or ASPP is deployed. These settings are exactly the same as the original authors use.

3.3 Differentiator

The siamese encoder-decoder will produce pixel-level semantic feature maps for the input image pairs. This provides a basic input with much less interference for the differentiator. At the uppermost layers of the two parallel encoder-decoder branches, we take the absolute difference of the two feature maps. This is the differencing process, and the formulation is as follows:

$$L_{i,j,k} = \sigma(|A_{i,j,k} - B_{i,j,k}|) \quad (1)$$

where L is the feature map produced by the differentiator, A and B are feature maps produced by the two decoders, and $\sigma(\cdot)$ refers to the softmax layer. k takes two values here: 0 and 1. Therefore, $L_{i,j,0}$ denotes the probability of pixel (i, j) being unchanged and $L_{i,j,1}$ denotes the probability of that pixel being changed.

Then we use the labeled change mask and the obtained prediction L to train the network in an end-to-end way with the cross-entropy loss.

In our model, the encoder and decoder components are completely decoupled, and hence they can be replaced by other CNN architectures and combined to form new networks. In our experiments, we will demonstrate this point in detail.

4 EXPERIMENTS

4.1 DataSet

We conduct experiments on two datasets: TSUNAMI and GSV. Both datasets consist of 100 pairs of 224 $\times$ 1024 street scene images. The TSUNAMI dataset contains images of tsunami-damaged areas of Japan, which are captured by a running vehicle on the street. The GSV dataset contains Google Street View images.

Image pairs in both datasets are coregistered beforehand by (Sakurada and Okatani, 2015). For each image pair, a binary image is provided as the ground truth, which indicates whether a change occurred for each corresponding pixel pair. Changes in these two datasets are defined as changes occurred on the surface of objects (the surface of buildings) and structural changes (appearing/disappearing objects). Therefore, grounds, skies, clouds, and illuminations are not regarded as changes. Further information about both of the two datasets can be found in (Sakurada and Okatani, 2015).

An example of changes such as the buildings and cars are shown in Fig. 1. Both the datasets contain many irrelevant visual differences between image pairs which meet our demands to train a robust model.

4.2 Configuration

Network structures, training policy, and parameter settings are detailed in the following paragraphs.

Fine-Tune. VGG16 and Resnet-101, the encoders in our model, are pretrained on ImageNet. FC, FC-F, ASPP, and ASPP-F stand for Fully Convolutional layer, Fully Convolutional layer with Fusion, Atrous Spatial Pyramid Pooling and Atrous Spatial Pyramid Pooling with Fusion respectively. Decoders are trained from scratch except for FC-F, which is fine-tuned from a trained FC, as FC-F trained from scratch is slow to converge and hard to outperform its FC counterpart.

Data Augmentation. Due to the insufficiency of data, data augmentation is necessary to train a decent model. We augment our data in the following three ways. First, images are left-right flipped randomly with a probability of 0.5. Second, images are rescaled randomly from 0.5 to 2.0 times the original size. Third, crop an 800 $\times$ 174 patch from the image produced in the second phase. Before the third phase, an image might be smaller than 800 $\times$ 174 if shrunk too much in the second phase. In this case, some pixels should be padded to the right and bottom of the image, in order to guarantee it's not smaller than 800 $\times$ 174. Afterward, these pixels will be ignored when the loss is calculated as they are useless for training.

Optimization. Our models are implemented using Tensorflow and trained on a single NVIDIA TITAN Xp. We use standard stochastic gradient descent with batch size of 8, momentum of 0.9 and weight decay of 0.0005 to train our models via back propagation. Each model is trained for 600 epochs except FC-F is fine-tuned on a pretrained FC for 1000 epochs (Data augmentation substantially reduces the risk of overfitting if trained for too many epochs). Initial learning rates for VGG16 and Resnet-101 are set to 0.001 and 0.007 respectively, with the exception of VGG16+FC-F setting to 0.0002. For each model, the learning rate is controlled by the polynomial learning rate policy:

$$\alpha_k = \alpha * (1 - \frac{k}{m})^p \tag{2}$$

where α_k is the learning rate at iteration k, α is the initial learning rate, k is the current iteration, m is the number of iterations to finish the training process and p is power mentioned above.

In our experiments, we train and evaluate our models via 5-fold cross-validation, i.e., 80 image pairs for training and 20 images for validation, the same as configured in (Sakurada and Okatani, 2015). We report two common metrics from semantic segmentation and machine learning: mean Intersection over Union (mean IoU) and F1 score. And It can be easily proved that $\mathrm{F1}/2 \leq \text{mean IoU} \leq \mathrm{F1}$.

4.3 Results and Discussion

Table 2 lists the performance of our models composed of different encoders and decoders. On TSUNAMI dataset, our best model VGG16+FC outperforms (Sakurada and Okatani, 2015)'s and (Alcantarilla et al.,

Table 2: Experiment results. *"no aug" means no data augmentation processes are carried out. The suffix *-F means low-level feature maps are fused. *"Dila" means dilated convolutional layers are used.

Model	TSUNAMI mean IoU	TSUNAMI F1 score	GSV mean IoU	GSV F1 score
(Sakurada and Okatani, 2015)	-	0.723	-	0.639
(Alcantarilla et al., 2018)	-	0.774	-	0.614
VGG16+FC (no aug*)	0.617	0.751	-	-
VGG16+FC	**0.707**	**0.819**	0.526	0.671
VGG16+FC-F*	0.662	0.788	0.466	0.613
VGG16+ASPP	0.579	0.718	0.409	0.516
VGG16+ASPP-F	0.618	0.752	0.512	0.662
Resnet-101+FC	0.597	0.740	0.531	0.675
Resnet-101+FC-F	0.583	0.722	0.509	0.654
Resnet-101+ASPP	0.477	0.635	0.496	0.642
Resnet-101+ASPP-F	0.569	0.719	0.520	0.661
Resnet-101+Dila*+ASPP	0.634	0.767	0.539	0.681
Resnet-101+Dila+ASPP-F	0.614	0.753	**0.545**	**0.697**

2018)'s by 0.096 and 0.045 respectively. On GSV dataset, our best model Resnet-101+Dila+ASPP-F outperforms (Sakurada and Okatani, 2015)'s and (Alcantarilla et al., 2018)'s by 0.058 and 0.083 respectively.

Our best results on TSUNAMI and GSV are achieved by VGG16+FC and Resnet-101+Dila+ASPP-F respectively. Results of GSV are much worse than those of TSUNAMI. We attribute it to the complexity of GSV. First, there are much more small objects in GSV than TSUNAMI, such as people, cars and trees. These small objects are hard to be identified by models that lose too much spatial information such as VGG16+FC. An example of these disappearing small objects is illustrated in Fig. 3. Note that cars in the middle of the scene and branches of trees have totally gone in the prediction of VGG+FC. Second, boundaries in GSV are sharper. How to refine boundary is a hard problem in semantic segmentation, and more smooth boundaries in the predictions of our models cause the F1 score to be lower than those in TSUNAMI.

Table 2 also shows the detailed performance of models constructed with different encoders and decoders.

FC and ASPP. ASPP outperforms FC only when dilated convolutional layers are present. ASPP captures multi-scale features using large dilated convolutional kernels. Thus, ASPP makes full of its advantages when the input feature maps are large. However, the image heights in our experiments are 174 and 224 in training and inference process respectively and will be reduced to 11 and 14. In this case, the size of feature maps is even smaller than the size of abovementioned dilated convolutional kernels, and leads to bad performance of ASPP in the absence of dilated kernels.

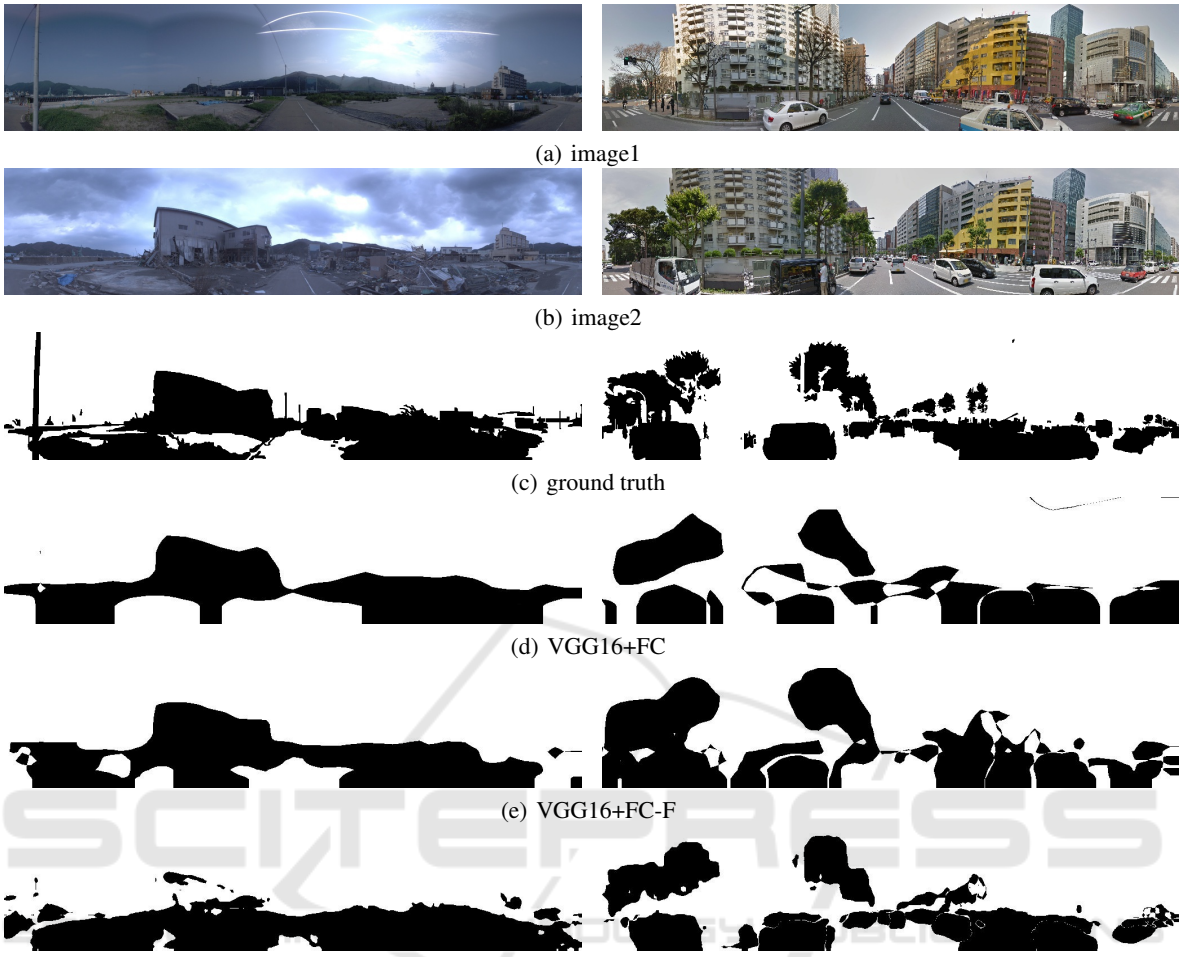
Fusion and Dilation. Fusion improves the performance of ASPP largely when dilated kernels are not used. When dilated kernels are used, the advantages of Fusion are not so obvious. Fusion introduces spatial information from low-level features and dilated kernels preserve low-level information. They perform the same function to some extent. So it explains why Fusion helps less when dilated kernels are present.

Another notable thing is that data augmentation is necessary for the robustness of our models, especially when the dataset is small. As Table 2 shows, VGG16+FC with augmentation obtains a much better result than VGG16+FC without augmentation. Fig. 4 partly explains the reason: model training without augmentation suffers from overfitting.

Basically, Resnet-101 combined with ASPP performs much better than VGG16 combined with FC in semantic segmentation on several datasets in almost all aspects. However, VGG16+FC soundly beats Resnet-101+Dila+ASPP-F on TSUNAMI as shown in Table 2, which is very counterintuitive. As we tracked the training and validation process, we found it suffer from overfitting. Resnet-101+Dila+ASPP-F is a complex model and good at recovering fine boundaries. On the one hand, the number of images in TSUNAMI is too small to properly fine-tune such a complex model. On the other hand, there are fewer objects in TSUNAMI than GSV and most objects are large in size. Although VGG16+FC cannot recover fine boundaries of objects, this doesn't affect the accuracy as severely as that in GSV.

5 CONCLUSIONS

In this paper, we have proposed a novel approach, called SEDS-CNN, for street-view change detection. The SEDS-CNN model is able to handle the irrelevant visual differences in change detection by introducing the encoder-decoder parts. VGG+FC gi-



(a) image1

(b) image2

(c) ground truth

(d) VGG16+FC

(e) VGG16+FC-F

(f) Resnet-101+Dila+ASPP-F

Figure 3: Illustration of our models on one TSUNAMI sample image pair and one GSV sample image pair. The left column comes from TSUNAMI and the right column comes from GSV.

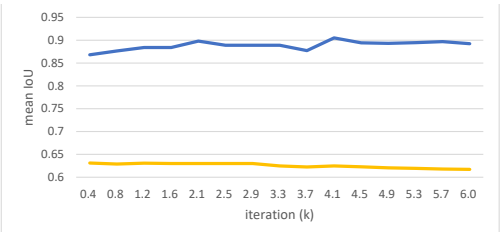


Figure 4: Training (80 pairs) and validation (20 pairs) mean IoU over iterations for VGG16+FC with no data augmentation. It's suffering from overfitting.

ves a decent result when the scenes are simple, objects are large and datasets are small, while Resnet-101+Dila+ASPP-F performs better when the opposite is true. Experiments show that techniques in semantic segmentaion benefit change detection.

The encoder and decoder parts are decoupled. It is flexible to choose various CNN architectures as encoders and decoders. Moreover, it is convenient to train the SEDS-CNN model in an end-to-end way. Experiments on TSUNAMI and GSV datasets demonstrate that the proposed SEDS-CNN model outperforms previous methods by a large margin.

ACKNOWLEDGEMENTS

This work was supported in part by the Natural Science Foundation of China under Grants U1435220 and 61503365, Youth Innovation Foundation of the 4th China High Resolution Earth Observation Conference under Grant GFZX04061502, and STS project of Fujian Province and Chinese Academy of Sciences under Grant 2018T3009.

REFERENCES

Alcantarilla, P. F., Stent, S., Ros, G., Arroyo, R., and Gherardi, R. (2018). Street-view change detection with deconvolutional networks. *Autonomous Robots*, 42(7):1301–1322.

Bruzzone, L. and Prieto, D. F. (2000). Automatic analysis of the difference image for unsupervised change detection. *IEEE Transactions on Geoscience and Remote sensing*, 38(3):1171–1182.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*.

Gong, M., Zhao, J., Liu, J., Miao, Q., and Jiao, L. (2016). Change detection in synthetic aperture radar images based on deep neural networks. *IEEE transactions on neural networks and learning systems*, 27(1):125–138.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Li, Y., Martinis, S., Plank, S., and Ludwig, R. (2018). An automatic change detection approach for rapid flood mapping in sentinel-1 sar data. *International Journal of Applied Earth Observation and Geoinformation*, 73:123–135.

Lin, T.-Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. (2017). Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Radke, R. J., Andra, S., Al-Kofahi, O., and Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE transactions on image processing*, 14(3):294–307.

Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. *arXiv preprint*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Sakurada, K. and Okatani, T. (2015). Change detection from a street image pair using cnn features and superpixel segmentation. In *BMVC*, pages 61–1.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, A. (1989). Review article digital change detection techniques using remotely-sensed data. *International journal of remote sensing*, 10(6):989–1003.

Tewkesbury, A. P., Comber, A. J., Tate, N. J., Lamb, A., and Fisher, P. F. (2015). A critical synthesis of remotely sensed optical image change detection techniques. *Remote Sensing of Environment*, 160:1–14.

Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890.