**Flood Level Detection based on Computer Vision Techniques**

*Machine Vision (16:332:561:01)*

|  |  |
| --- | --- |
| Professor: | Yuqian Zhang |
| Members: | Chong Di |
|  | Jiahao Xia(jx198) |

***Date: 2020-11-01-2020-12-04***

**Contents**

1. Introduction
2. Related Work

2.1 Semantic Segmentation

2.2 Image Matching

2.3 Camera Pose Estimation

1. Method
2. Datasets
3. Experiments
4. Conclusions
5. References
6. **Introduction**

Flood exposure is increasing in coastal cities owing to growing populations and assets, the changing climate, and subsidence [1]. By the year 2100, between 0.2–4.6% of the global population and 0.3–9.3% of global gross domestic product may be exposed to coastal flooding if no adaptation occurs. Flood protection and awareness have continued to rise on the political agenda over the last decade. Management of flood risk relies on statistical and hydrodynamic modeling to delineate populations and assets exposed to flooding, anticipate and monetize the consequences of flooding, and develop cost effective and socially robust interventions including infrastructure projects, insurance programs, land use and building code policy changes and emergency preparedness and response measures. Among these, the flood level estimation plays an important role in addressing the risks and supporting emergency plan operations [2].

To reduce the risk of human fatalities, it is important to have accurate flood maps that can properly guide rescue operations. To build such maps it is necessary to retrieve real-time scattered information about the flood-water level from the disaster area. Classical monitoring systems include stream gauge, remote sensing, and field data collection. These methods, however, reveal several limitations. For instance, remote sensing data from satellites, though being rather inexpensive, does not provide real-time access during a disaster since the revisit cycle of the satellite is usually too large. On field data collection is instead usually expensive and dangerous as it requires to inspect the disaster area. A viable alternative source of information in this case comes from social media platforms. People located in areas affected by the flood often share pictures describing the situation. These images have the advantage to be cheap and available in real-time directly from the flooded region[3-5].

1. **Related Work**

**2.1 Semantic Segmentation**

During the long history of computer vision, one of the grand challenges has been semantic segmentation which is the ability to segment an unknown image into different parts and objects. Furthermore, segmentation is even deeper than object recognition because recognition is not necessary for segmentation. Specifically, humans can perform image segmentation without even knowing what the objects are (for example, in satellite imagery or medical X-ray scans, there may be several objects which are unknown, but they can still be segmented within the image typically for further investigation). Performing segmentation without knowing the exact identity of all objects in the scene is an important part of our visual understanding process which can give us a powerful model to understand the world and also be used to improve or augment existing computer vision techniques. In the following summary, we introduce classic deep neural networks used in semantic segmentation.

1. U-Net [6]

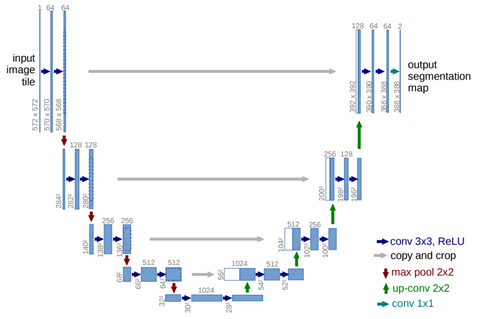


Figure 1. Illustration of U-Net (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

U-Net, evolved from the traditional convolutional neural network, was first designed and applied in 2015 to process biomedical images. As a general convolutional neural network focuses its task on image classification, where input is an image and output is one label, but in biomedical cases, it requires us not only to distinguish whether there is a disease, but also to localize the area of abnormality. U-Net is dedicated to solving this problem. The reason it can localize and distinguish borders is by doing classification on every pixel, so the input and output share the same size.

The U-Net architecture is illustrated in Figure 1. It is symmetric and consists of two major parts: the left part is called contracting path, which is constituted by the general convolutional process; the right part is expansive path, which is constituted by transposed 2d convolutional layers. The network is strong enough to do good prediction based on even few data sets by using excessive data augmentation techniques. There are many applications of image segmentation using U-Net, such as autonomous vehicles, biomedical image diagnosis, geo-sensing and precision agriculture.

1. SegNet [7]

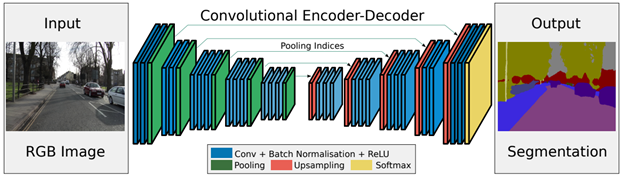


Figure 2. Illustration of SegNet. There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification.

As Figure 2 indicates, SegNet has an encoder network and a corresponding decoder network, followed by a final pixelwise classification layer. At the encoder, convolutions and max pooling are performed. There are 13 convolutional layers from VGG-16, but the original fully connected layers are discarded. While doing 2×2 max pooling, the corresponding max pooling indices (locations) are stored.

At the decoder, upsampling and convolutions are performed. At the end, there is softmax classifier for each pixel. During upsampling, the max pooling indices at the corresponding encoder layer are recalled to upsample. Finally, a K-class softmax classifier is used to predict the class for each pixel. The contribution of SegNet is that it Uses a novel technique to upsample encoder output which involves storing the max-pooling indices used in pooling layer. This gives reasonably good performance and is space efficient. Another point is that the VGG16 with only forward connections and non-trainable layers is used as the encoder and this leads to very less parameters.

Some applications of SegNet include autonomous driving, scene understanding, etc. Direct adoption of classification networks for pixel wise segmentation yields poor results mainly because max-pooling and subsampling reduce feature map resolution and hence output resolution is reduced.

1. Fully Convolutional Networks (FCNs) [8]

The basic idea behind a fully convolutional network is that it is “fully convolutional”, that is, all its layers are convolutional layers. FCNs don’t have any of the fully-connected layers at the end, which are typically used for classification. Instead, FCNs use convolutional layers to classify each pixel in the image. So the final output layer will be the same height and width as the input image, but the number of channels will be equal to the number of classes. If we’re classifying each pixel as one of fifteen different classes, then the final output layer will be height\*width\*15.

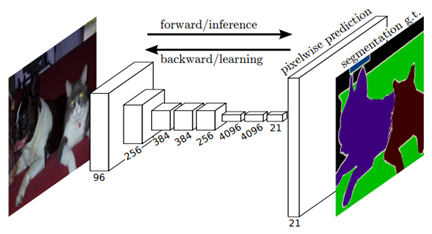


Figure 3. Illustration of FCNs. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

A logistical hurdle to overcome in FCNs is that the intermediate layers typically get smaller and smaller (although often deeper), as striding and pooling reduce the height and width dimensions of the tensors. FCNs use “deconvolutions”, or essentially backwards convolutions, to upsample the intermediate tensors so that they match the width and height of the original input image.

The authors had success converting canonical networks like AlexNet, VGG, and GoogLeNet into FCNs by replacing their final layers. But there was a consistent problem, which was that upsampling from the final convolutional tensor seemed to be inaccurate. Too much spatial information had been lost by all the downsampling in the network. So they combined upsampling from that final intermediate tensor with upsampling from earlier tensors, to get more precise spatial information.

1. DeepLab [9]

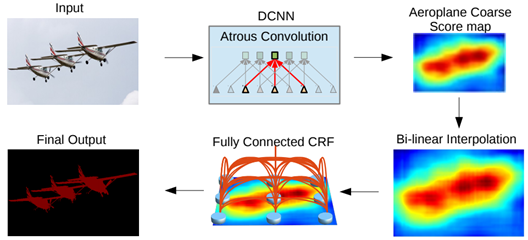


Figure 4. Model Illustration. A Deep Convolutional Neural Network such as VGG-16 or ResNet-101 is employed in a fully convolutional fashion, using atrous convolution to reduce the degree of signal downsampling (from 32x down 8x). A bilinear interpolation stage enlarges the feature maps to the original image resolution. A fully connected CRF is then applied to refine the segmentation result and better capture the object boundaries.

DeepLab is a state-of-art deep learning model for semantic image segmentation, where the goal is to assign semantic labels to every pixel in the input image. Current implementation includes the following versions:

• DeepLabv1 : They use atrous convolution to explicitly control the resolution at which feature responses are computed within Deep Convolutional Neural Networks.

• DeepLabv2 : They use atrous spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales with filters at multiple sampling rates and effective fields-of-views.

• DeepLabv3 : They augment the ASPP module with image-level feature to capture longer range information. They also include batch normalization parameters to facilitate the training. In particular, They applying atrous convolution to extract output features at different output strides during training and evaluation, which efficiently enables training BN at output stride = 16 and attains a high performance at output stride = 8 during evaluation.

• DeepLabv3+: They extend DeepLabv3 to include a simple yet effective decoder module to refine the segmentation results especially along object boundaries. Furthermore, in this encoder-decoder structure one can arbitrarily control the resolution of extracted encoder features by atrous convolution to trade-off precision and runtime.

1. PSPNet [10]

By using Pyramid Pooling Module, with different-region-based context aggregated, PSPNet surpasses state-of-the-art approaches such as FCN and DeepLab, and PSPNet finally got the champion of ImageNet Scene Parsing Challenge 2016 and Arrived 1st place on PASCAL VOC 2012 & Cityscapes datasets at that moment.

As illustrated in Figure 5, at the extracting feature map stage, ResNet is used with dilated network strategy. For the pyramid pooling module, sub-region average pooling is performed for each feature map and then 1×1 convolution is performed for each pooled feature map to reduce the context representation. Next, bilinear interpolation is performed to up-sample each low-dimension feature map to have the same size as the original feature map and all different levels of upsampled feature maps are concatenated with the original feature map. These feature maps are fused as global prior. At the final stage, it is followed by a convolution layer to generate the final prediction map.

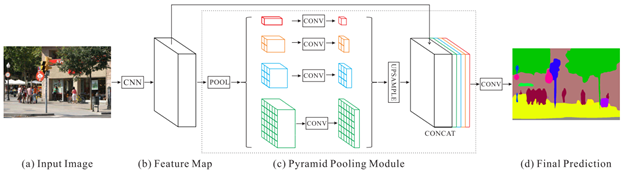


Figure 5. Overview of PSPNet. Given an input image (a), we first use CNN to get the feature map of the last convolutional layer (b), then a pyramid parsing module is applied to harvest different sub-region representations, followed by upsampling and concatenation layers to form the final feature representation, which carries both local and global context information in (c). Finally, the representation is fed into a convolution layer to get the final per-pixel prediction (d).

**2.2 Image Matching**

The Scale-invariant feature transform (SIFT) image matching method that employs feature vectors, also known as keypoints to describe a local area on a target image. This supervised approach reduces the content of the reference image to a set of feature points that can also be found on a new image from another viewpoint. In the description stage, feature detector is usually created as a non-overlapping grid of a specific size (i.e. 4 ⨉ 4, 16 ⨉ 16) called local area and keypoints are identified as the extrema by pyramiding multiple Gaussian-blurred feature detectors.

SURF

BRIEF

**2.3 Camera Pose Estimation**

Camera pose determination from known 3D space points is called the perspective-n-point problem, namely, the PnP problem. When n = 1,2, there are no solutions for PnP problems because they are under constraints. When n ≥ 6, PnP problems are linear. When n =3, 4, 5, the original equations of PnP problems are usually nonlinear. The PnP problem dated from 1841 to 1903. Researchers conclude that the P3P problem has at most four solutions and the P4P problem has a unique solution in general. The PnP problem is also the key relocalization for SLAM.

The methods to solve PnP problems with n =3, 4, 5 focus on two aspects. One aspect studies the solution numbers or multisolution geometric configuration of the nonlinear problems. The other aspect studies eliminations or other solving methods for camera poses.

In this project, we focus on PnP problems with n ≥6. When n≥6, PnP problems are linear and studies on them focus on two aspects. One aspect studies efficient optimizations for camera poses from smaller number of points. The other aspect studies fast camera localization from large data. The popularly used DLT method sometimes fails to give reliable camera parameter estimation. It is therefore important to detect the unreliability and provide the corresponding solutions. Based on a complete framework of invariance for six points, Wu et al. construct two evaluation functions to detect the unreliability. The two evaluation functions do not involve any computations for the camera projective matrix or optical center and thus are efficient to perform the detection. Then, the guidelines corresponding to the different detection results are presented. In particular, a filtering RANSAC method to remove the detected unreliable points is provided. The filtering RANSAC proves to be successful in removing the unreliable points even if these points are of a large proportion [11]. In order to realize fast camera pose estimation on large data, Sattler et al. derived a direct matching framework based on visual vocabulary quantization and a prioritized correspondence search with known large-scale 3D models of urban scenes [12]. Their results indicate that the framework efficiently handles large datasets and outperforms current state-of-the-art methods.

1. **Method**

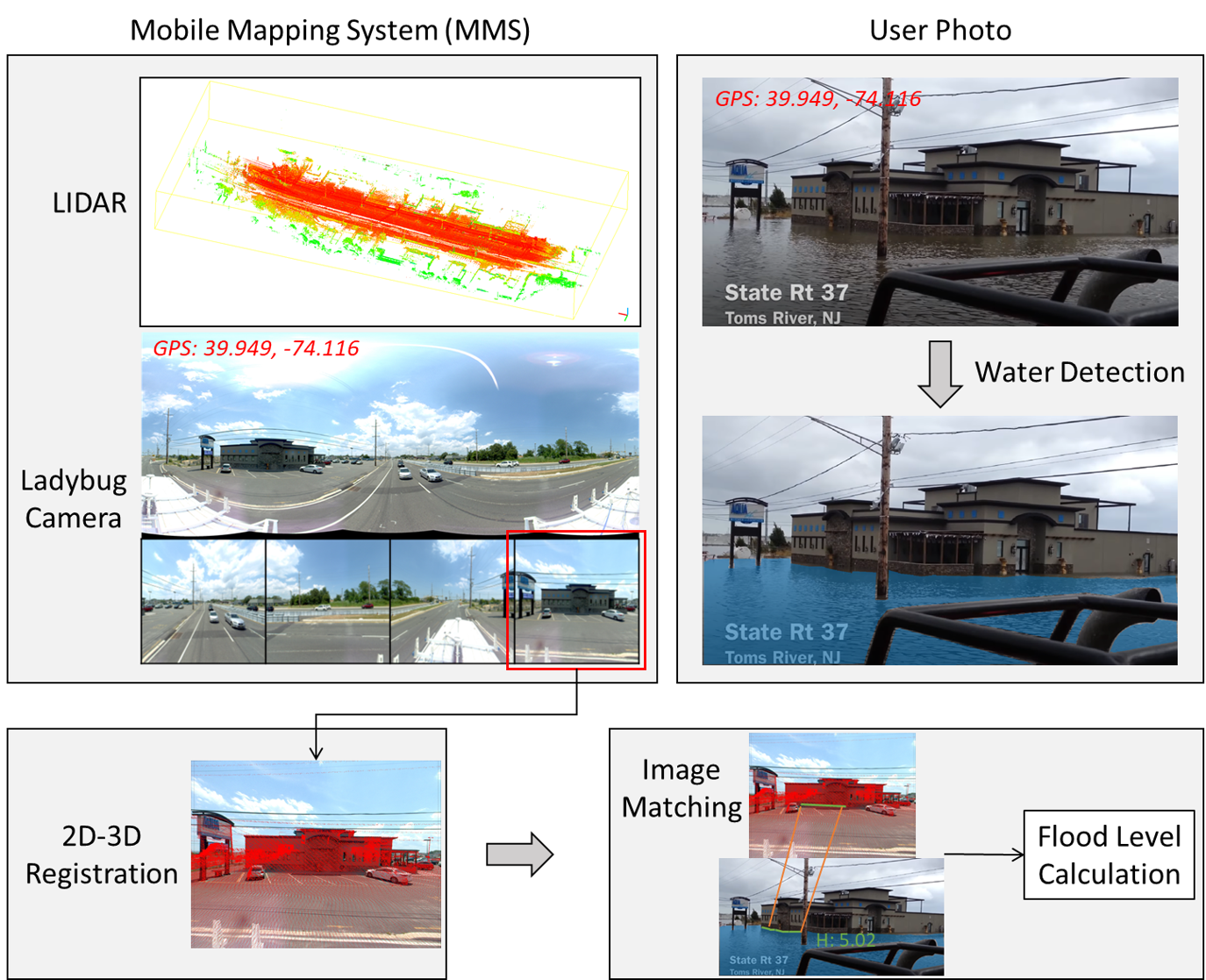


Figure 6. The flowchart of our proposed method. The mobile mapping system can collect point cloud and panoramic images, and the images are registered with point cloud. We use the semantic segmentation model to extract water from the user photos. The flood level is calculated based on the matched 3D point cloud.

1. **Datasets**
2. Point cloud data

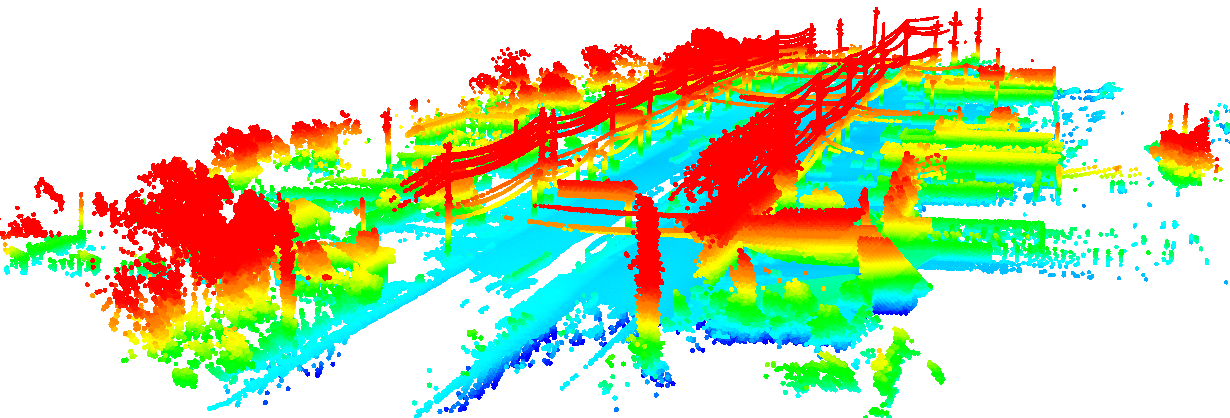


Figure 7. Illustration of the point cloud.

1. Panoramic images



Figure 8. Illustration of panoramic image

1. Flood images

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Figure 9. Illustration of flood images

1. **Experiments**
2. **Conclusions**
3. **References**

1. Hallegatte, S., et al., Future flood losses in major coastal cities. 2013. 3(9): p. 802-806.

2. Moftakhari, H., et al., Linking statistical and hydrodynamic modeling for compound flood hazard assessment in tidal channels and estuaries. Advances in Water Resources, 2019. 128: p. 28-38.

3. Chaudhary, P., et al., Flood-water level estimation from social media images. 2019. 4(2/W5): p. 5-12.

4. Pan, J., et al., Deep learning-based unmanned surveillance systems for observing water levels. 2018. 6: p. 73561-73571.

5. Vallimeena, P., B.B. Nair, and S.N. Rao. Machine Vision Based Flood Depth Estimation Using Crowdsourced Images of Humans. in 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). 2018. IEEE.

6. Ronneberger, O., P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. in International Conference on Medical image computing and computer-assisted intervention. 2015. Springer.

7. Badrinarayanan, V., et al., Segnet: A deep convolutional encoder-decoder architecture for image segmentation. 2017. 39(12): p. 2481-2495.

8. Long, J., E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

9. Chen, L.-C., et al., Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. 2017. 40(4): p. 834-848.

10. Zhao, H., et al. Pyramid scene parsing network. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

11. Wu, Yihong, Youfu Li, and Zhanyi Hu. "Detecting and handling unreliable points for camera parameter estimation." International Journal of Computer Vision 79.2 (2008): 209-223.

12. Sattler, Torsten, Bastian Leibe, and Leif Kobbelt. "Fast image-based localization using direct 2d-to-3d matching." 2011 International Conference on Computer Vision. IEEE, 2011.