

Flood Water Segmentation for Crowdsourced Images

John K. Nguyen^{a,*}, Barbara S. Minsker^b

^a*Illinois Informatics Institute, University of Illinois at Urbana-Champaign*

^b*Civil & Environmental Engineering Department, Southern Methodist University*

Abstract

The proliferation of camera-equiped mobile devices have led to a new source of information for flood research. In-situ photographs captured by people provides information at the local level that remotely sensed images fail to capture. Applications of crowdsourced images to flood research requires understanding the content of the image without the need for user-input. This paper addresses the problem of how to automatically determine a flooded and non-flood region in crowdsourced images. Previous works required two images taken at similar angle and perspective of the location when it is flooded and when it is not flooded. We examine three different algorithms that is able to perform segmentation using a single flood image without these assumptions. The performance of each algorithm is evaluated on a collection of crowdsourced flood images. We show that it is possible to accurately segment a flood image using just a single image.

Keywords: flood water segmentation, crowdsourcing, image processing, computer vision

1. Introduction

The use of crowdsourced information in floods is a very recent active research area. In [1], the quality of geographic information gathered from an flood-affected population was compared to hydraulic models. The authors found that the affected population was able to estimate flood water level as accurate as hydraulic models. Another study on the quality of crowdsourced information for flood research was given by [2]. In a case study on flood risk assessment for Sao Carlos, Brazil watershed, the authors found that the volunteered geographic information is comparable to data collected from sensors. The usefulness of crowdsourced geographic information have led to the development of cyber-infrastructure systems and decision-support tools with crowdsourcing capabilities [3, 4, 5]. These tools allow the public and emergency responders to input information such as severity classification, number of fatalities, and flood duration.

As more and more people become equipped with mobile phones with camera capability, photographs and videos are becoming a very common medium of information. Crowdsourced photographs pose new opportunities for flood research which have traditionally relied on remotely sensed radar and satellite images for flood risk assessment [6, 7], flood mapping [8], flood monitoring [9, 10], and flood damage assessment [11, 12]. While remotely sensed images provides wholistic analysis of an entire region, they do not capture

*Corresponding author

information at the local level. For example, information such as flood depth is more accurate from an in-situ image than depth sensors located far from the inundated area [13].

The first major usage of crowdsourced flood images for flood research is the FP7 SPACE Project GEO-PICTURES, a collaborative effort between the United Nations (UNOSAT) and AnsuR Technologies [14].
20 Through a mobile-app called UN-ASIGN, crowdsourced, geo-tagged photos captured on smartphones were used in conjunction with radar images to improve flood assessment. Using the data collected from the 2011 Thailand floods, the Thai government and Asian Disaster Prevention Center developed better emergency management response.

As crowdsourced flood images become incorporated into future decision-support tools, there is a need to
25 automatically process images. One of the fundamental problems in working with crowdsourced flood images is *flood water segmentation*, the partition of the flooded region from the rest of the image. To automate such a task poses a major challenge due to the lack of standardization, quality assurance, or quality control. Consider remotely sensed LiDAR images which must follow the base specifications for quality dictated by U.S. Geological Survey (USGS) [15]. Standardized images such as LiDAR makes the application of
30 image processing much easier as there is limited variability in how the image was captured and its content. Crowdsourced images, however, could be taken from different elevations with varying angles, be of extremely poor or high quality, and may contain numerous visible objects (e.g., people, trees, cars, etc.). Algorithms to segment the flood water from crowdsourced images must be able to work with the gamut of information contained in each image.

35 2. Related Work & Contributions

There have been several previous work in segmenting flood water from images for the purpose of water level detection. The common assumption of these works is that all images taken are at a consistent perspective and angle from a pre-calibrated camera. The paper by [16] utilize the accumulated histogram which shows changes in a sequence of images. To separate the water and non-water region, the authors used a bandpass
40 filter is used to remove noise and a set of decision rules based on the pulse histograms of each image. Another image differencing approach was used in [17] to detect changes in water level. Using a reference image and a acquisition image from a pre-calibrated camera, the change between the two images was determined by comparing the change in the Y-axis profile of the Sobel edge image. We direct the readers to [13] for a comprehensive survey of other image differencing approaches to this problem.

45 The use of crowdsourced flood images for water level detection was introduced in [18]. Participants captured and upload a geo-tagged image of a flood containing submerged static structures such as a building and lampposts. Then, using a previously stored reference image, a flood line is estimated based on invariant feature transform (SIFT). The segmentation of flood and non-flood regions in crowdsourced flood images was studied more generally in [19]. Using a non-flood and a flood image of the same location, the authors

50 determine the flood region via image subtraction based on the specularity (i.e., reflection) of the flood water between the two images.

Many of the works in water level detection presented above is not applicable for all flood images as they assume pre-calibrated cameras. Also, within the context of crowdsourcing, it is not possible to apply image differencing approaches in general as a non-flood image of the exact location might not be available or might
55 not have been taken with the same conditions as the flood image. In this work, we relax the assumption of a pre-calibrated camera and the need for two images.

The main contribution of this paper is to show that we can perform flood water segmentation from just a single crowdsourced image. We make no assumptions on how the image was captured and its content. We propose three algorithms used in the image processing and computer vision to this problem. The
60 performance of the algorithms is evaluated on a real dataset of crowdsource images taken from different flood events. Discussions on how to improve the performance of both algorithms as well as future research using crowdsourced flood images will be also given.

3. Crowdsourced Flood Image Dataset

The dataset used in this work consists of 25 publicly available in-situ flood images taken from different
65 flooding events. To simulate real-world application, these images are of different dimensions, taken with different perspectives and angle, and contain high variability in content. Some characterization of the images includes:

- **Slanted or non-linear water horizon:** All images either contains a slanted water horizon or a non-linear horizon.
 - **Ambiguous water horizon:** The water horizon in a flood image may be ambiguous or there can be multiple horizons.
 - **Lack of color features:** As floods occur during or after a storm, the color information of the image might be limited, making it difficult to classify if a pixel belongs to the water region or the sky region.
 - **Present of objects:** There might exist objects on the water horizon such as cars, people, and debris.
- 75 Most of the images in our dataset will contain a combination of these characterizations. A sample of the images used is provided in Figure 1.

4. Methodologies

We present two algorithms which are applicable for the flood water segmentation problem. The first algorithm is found in the domain of image processing where we segment an image based on transformations
80 and local operations at the pixel level. These types of algorithms make little to no assumptions about the



(a)



(b)



(c)



(d)



(e)



(f)

Figure 1: Examples of images contained in the dataset.

image content. The algorithm is found in the domain of computer vision. The goal of these computer vision algorithms is to extract information from the image upon learning features about the image content. As such, a large database of images is required to train these models to produce accurate results.

4.1. Horizon Line Detection for Flood Images

The segmentation of flood water is similar to the horizon line detection (HLD) problem. The goal of HLD is to segment an image into sky and non-sky (ground, water, etc.) regions. There are several applications of HLD, including visual geo-localization [20], labeling mountain/desert landscape images [21, 22], control of unmanned aerial vehicles [23], and road recognition for autonomous vehicles [24]. While there have been some applications of HLD to water-based images such as marine vehicle detection [25], there have been no specific application of HLD to flood images. Research in HLD can be dichotomize as either (1) edge-based or (2) region-based.

4.1.1. Algorithm 1: Edged-based HLD

A popular HLD algorithm is to perform edge detection and Hough transformation [26, 27]. An image is first pre-processed by changing each pixel to only carry luminosity information (i.e., grayscaling) and reducing noise by passing a low-scale filter. Edge detection using a Canny [28] or Sobel filter [29] is then performed on the image. The resulting edge image is transformed into a Hough map which contains curves representing detected edges relative to its angle and its distance from a center. The peaks of a Hough map represents linear edges found in the image. If we assume that the flood water horizon is the most prominent linear line contained in the image, we can select the highest peak in the Hough map and mapping it back to the original image for segmentation. In Figure 2, we apply edge detection with Hough transformation for a flood image with a slanted water horizon. We first take the input image, grayscale it, and determine the Canny edges contained in the image. We perform a Hough transformation to extract the edge features of the image and select the curve with the highest peak. As the figure illustrate, we were able to accurately find the horizon line even with a slanted water horizon and the presence of two objects.

Note that edged-based HLD is very elementary and not applicable for most images as it assumes the horizon line is linear. As it is simply a horizontal partitioning of an image, HLD will not work very well for images with non-linear horizon. In addition, the flood water horizon might not be the most prominent line (i.e., the maximum peak might not correspond to the flood water horizon) which will cause the algorithm to incorrectly segment.

4.1.2. Algorithm 2: Flood Segmentation via SLIC

Current state-of-the-art algorithms in image processing uses *superpixels*, an image patch better aligned with intensity edges than rectangular pixel grids. A very efficient and popular segmentation algorithm of superpixels called *Simple Linear Iterative Clustering* (SLIC) was presented in [30]. Given the number of clusters and weighing between pixel space and intensity, SLIC uses the popular k -means to segment an

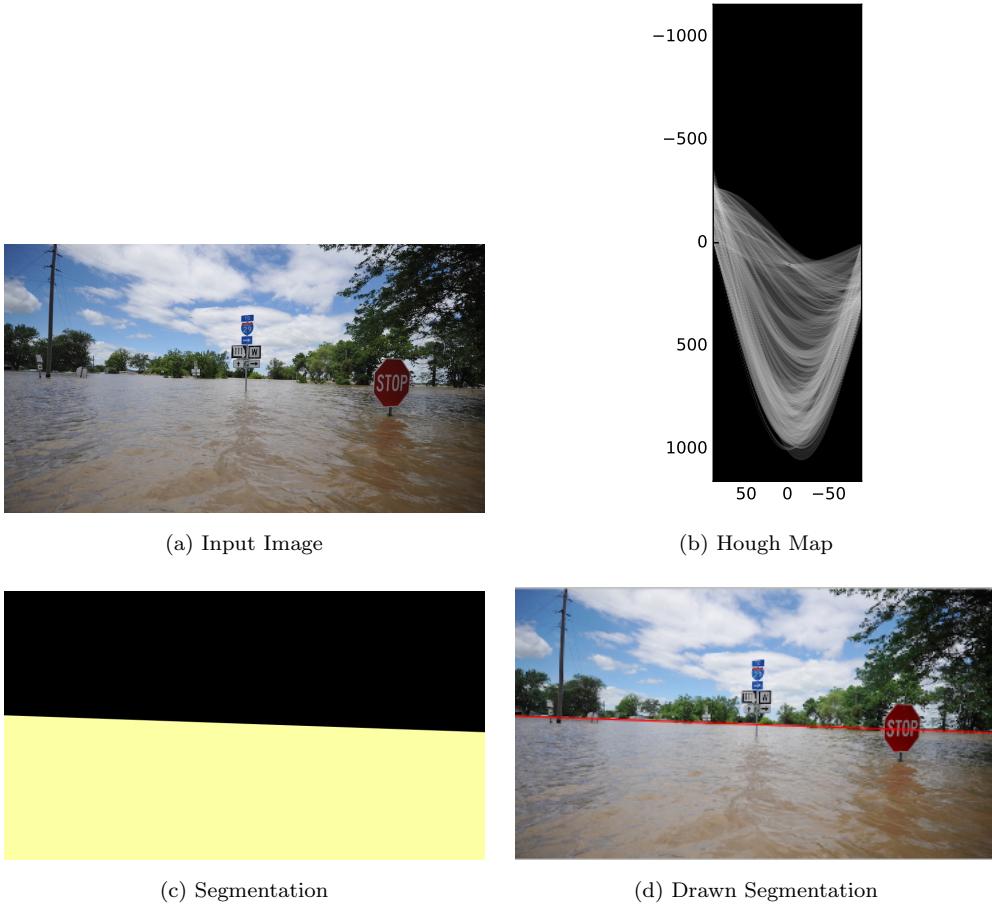


Figure 2: Example of edge detection with Hough transformation for linear water horizon line.

115 image in a given color space based on the color and space proximity of a superpixel. A version of SLIC called SLICO where the weights between color and space are chosen adaptively was also presented in [30].

We implement SLIC with $k = 2$ in Figure 3 for an image containing a nonlinear water horizon and different objects in the water. Although not perfectly accurate, we see that the algorithm was able to reasonably segment the flood region and non-flood region as two clusters. SLIC works reasonably well for 120 images with multiple or ambiguous flood water horizon and images with a large number of objects. It is also efficient and scalable for large image or a collection of images.

4.2. Deep Learning for Flood Images

The current state-of-the-art algorithms for segmentation in computer vision uses *deep learning* in the form of *convolutional neural networks* (CNNs) [31, 32, 33]. Unlike the traditional machine learning approach 125 which requires manual feature extraction before application of learning, CNNs employs *end-to-end learning* where features are trained jointly with the classification task.

CNNs enable us to extend the image segmentation to *semantic segmentation*. Recall that our image

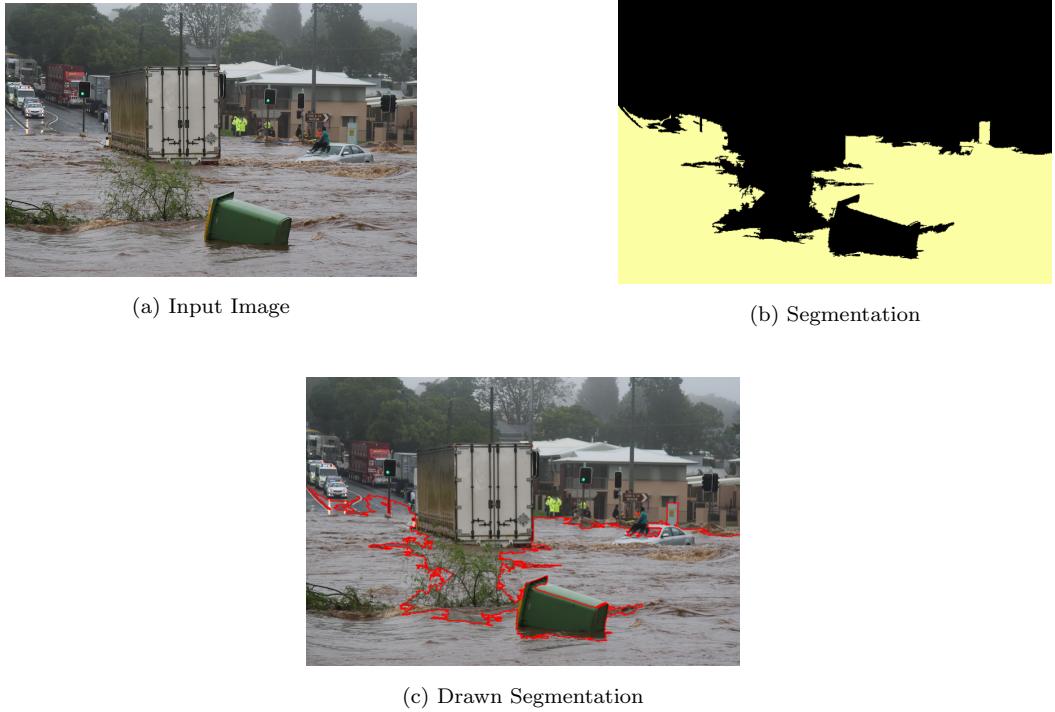


Figure 3: SLIC using 2 clusters and compactness of 0.1.

segmentation problem involves partition of an image into a flood region and a non-flood region. The two algorithms we had presented performs this segmentation but the resulting image contains no information on what these segmented parts actually represents. That is, we know which part consists of flood water through manual inspection only. Semantic segmentation involves the partition of an image into semantically meaningful parts with class labels on each parts.

Training a CNNs for semantic segmentation requires a large amount of data and computational power. However, training from scratch is often not necessary as one can employ *transfer learning* (also referred to as *fine-tuning*), improving a learner from one domain by transferring information from a related domain [34]. We will present a pre-trained CNN model called *fully convolutional neural networks* (FCN) that is applicable to our problem proposed in the works of [32]. Here, “fully” indicates that the neural network is composed of convolutional layers are without any fully-connected layers or multilayer perception usually found at the end of the network.

140 4.2.1. Algorithm 3: Flood Segmentation via FCN

We use the weights of the FCN model trained on the PASCAL-Context dataset which consists of over 400 labels for 10,103 images [35]. The network consists of 18 convolution layers, 5 pooling layers and 15 rectified linear units to apply the activation function $f(x) = \max(0, x)$ after each convolution. The output is a segmented image with scores associated with each of the 400 labels.

145 We take the score associated with the label “water” and compute the mean of the label score for each

region. Each pixel with a water score greater than or equal to the mean of the label is classified as a flood pixel. Note, while we can define any percentile of scores to classify, we found that the mean score works best as anything higher results in a high number of false positives and anything lower results in a high number of false negatives.

¹⁵⁰ **5. Evaluation**

5.1. Metrics

By far the best algorithm is FCN which has a mean pixel accuracy of 91% and a mean IU score of 84%.

The output of each algorithm is compared to the labeled image using two metrics: (1) pixel accuracy and (2) mean intersection over union (IU) score (also known as the Jaccard index). Pixel accuracy is simply the number of pixels correctly labeled divided the total number of pixels in the image. The IU score is the number of true positives divided by the sum of true positives, false positives and false negatives over each class. The latter metric is often the standard in the computer vision literature.

More formally, define n_{ij} be the pixel of class i predicted to belong to class j and $t_i = \sum_j n_{ij}$ be the total number of pixels of class i . The metrics are given as follows:

- ¹⁶⁰
1. Pixel Accuracy = $\frac{\sum_i n_{ii}}{\sum_i t_i}$
 2. Mean IU Score = $\frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}$

5.2. Result

The mean pixel accuracy and IU score over the entire crowdsource flood image dataset is given in Table 1. We see that simply using Hough transformation for image partitioning correctly classified approximately 80% of the pixels which is slightly better than SLIC. However, in considering the mean IU score where there is a greater penalty for misclassification, SLIC performs better than Hough. While the overall results

Table 1: Comparison of result on flood image dataset.

Algorithm	Mean Pixel Accuracy	Mean IU Score
Hough	0.7968	0.6696
SLIC	0.7828	0.6899
FCN	0.9125	0.8415

suggests that Hough and SLIC are comparable, if we examine specific cases, the difference between the two are very pronounced. Consider the example is given in Figure 4 where we see that SLIC gives a much better segmentation when the most prominent line does not correspond to the flood water horizon.

¹⁷⁰ Although the pixel accuracy for both Hough and SLIC are very high, the overall IU score suggests that they are not that reliable. By far the best algorithm is FCN which has a mean pixel accuracy of 91% and a mean IU score of 84%. The major advantage of FCN is that it is able to segment objects contained in the

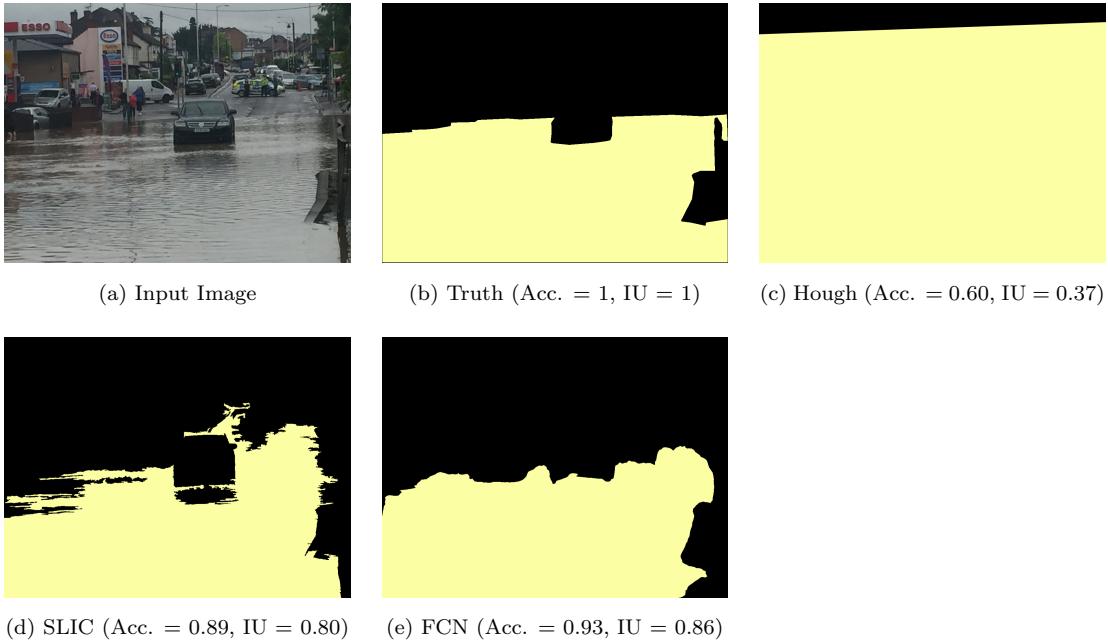


Figure 4: Example result of each algorithm for a test image.

flood better than SLIC. We show an example of such case in Figure 5 where FCN was able to segment the raft, dog and human from the flood water.

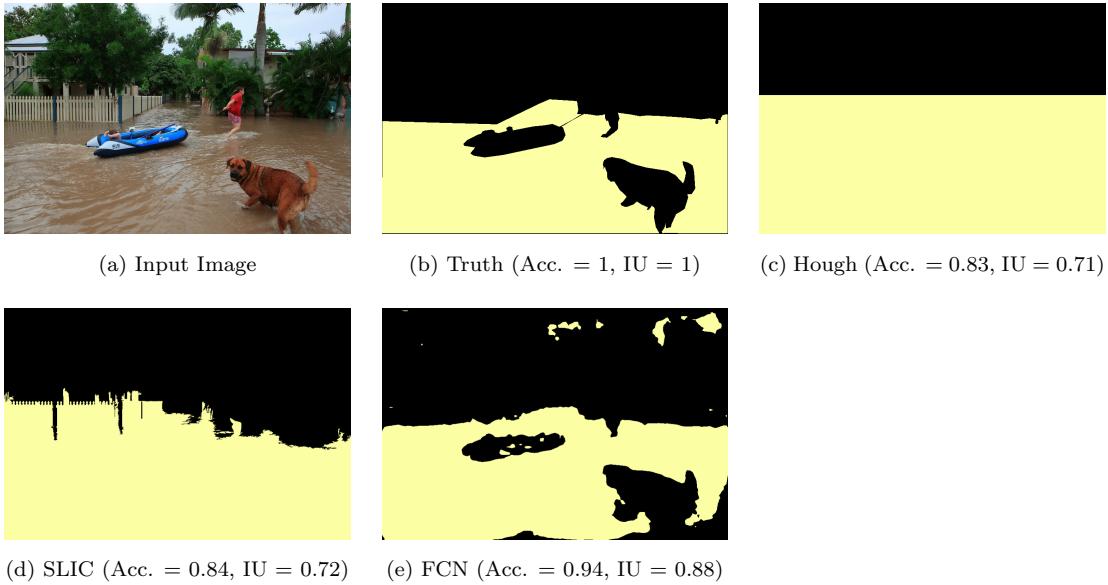


Figure 5: Example result of each algorithm for a test image.

175 *5.3. Drawbacks of Hough and SLIC*

Hough transformation works very well for flood images where the horizon line is linear and prominent. If there are other prominent lines contained in the picture, there will multiple peaks in the Hough map and the water horizon is not guaranteed to correspond to the highest peak. In Figure 6, the most prominent line corresponds to the roof of a building so the water horizon was not captured. While there are some advanced
180 edge detection techniques which can handle non-linear horizons, most image processing techniques lacks the ability to segment objects from flood water.



Figure 6: Hough transformation fails to capture the water horizon.

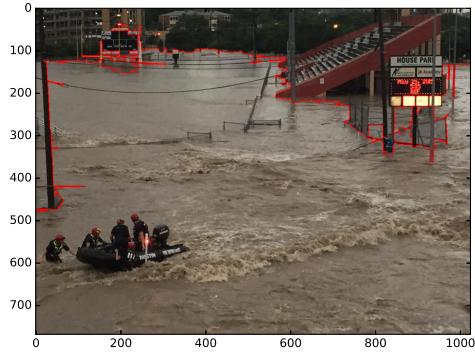
The main problem with unsupervised methods like SLIC is that it is not clear how we set the number of clusters. We can have vastly different segments or even no segments at all depending on the value of k . Another problem is the presence of specularity (i.e., reflections) in an image. SLIC tends to cluster pixels
185 found in reflective regions of the image.

In Figure 7(a), we see that $k = 3$ provides a reasonable segmentation for the image. However, when we set $k = 4$ in Figure 7(b), the segmentation changes greatly to something highly inaccurate. For a different input image shown in Figure 7(c), $k = 3$ does even not provide any segments as every pixel is grouped into the same cluster. When we increase $k = 4$ in Figure 7(d), we see that specularity region in the flood water
190 becomes a cluster. While it is possible to improve segmentation accuracy through specularity removal, such task is non-trivial.

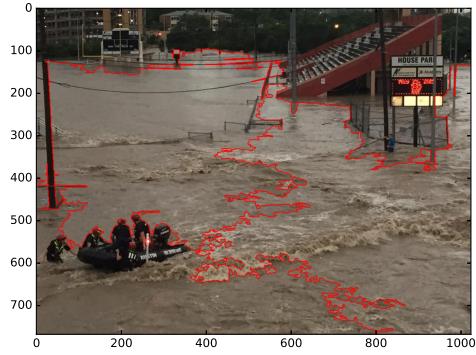
If we assume the image contains no specularity, then SLICO with k between 2 to 3 will give fairly good results. However, our goal is to be able to perform segmentation without such assumption. In addition, we would like a parameter-free model which would work for all flood images.

195 **6. Conclusion**

This study present several different approaches for flood water segmentation for crowdsourced images. We found that using a pre-trained convolutional neural networks, we showed that it is possible to perform parameter-free flood water segmentation using just a single image without any stringent assumptions on



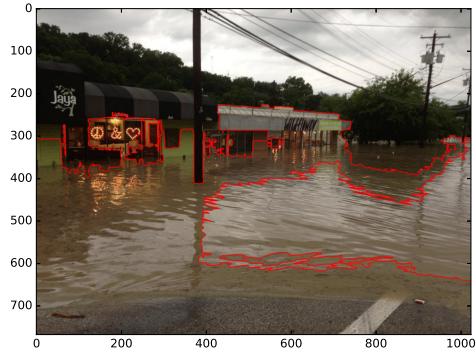
(a) SLIC with $k = 3$



(b) SLIC with $k = 4$



(c) SLIC with $k = 3$



(d) SLIC with $k = 4$

Figure 7: SLIC segments is highly dependent on the defined number of clusters and the presence of specularity.

image content. The model's accuracy is limited due to the lack of a labeled database of flood images. A
200 natural extension is to perform fine tuning or transfer learning of the model on a large, labeled flood image dataset. It would also be of interest to examine all the contents of an image and see if we can use such information for flood damage assessment and flood monitoring.

References

- [1] K. Poser, H. Kreibich, D. Dransch, Assessing volunteered geographic information for rapid flood damage estimation, in: 12th AGILE International Conference on Geographic Information Science, 2009.
205
- [2] R. de Brito Moreira, L. Degrossi, J. de Albuquerque, An experimental evaluation of a crowdsourcing-based approach for flood risk management, in: 12th Workshop on Experimental Software Engineering, 2015.
- [3] Z. Wan, Y. Hong, S. Khan, J. Gourley, Z. Flamig, D. Kirschbaum, G. Tang, A cloud-based global flood disaster community cyber-infrastructure: Development and demonstration, Environmental Modelling and Software 58 (86-94).
210

- [4] D. Lieske, Coping with climate change: The role of spatial decision support tools in facilitating community adaptation, *Environmental Modelling and Software* 68 (98-109).
- [5] F. E. Horita, J. P. Alburquerque, L. C. Degrossi, E. M. Mendiondo, J. Ueyama, Geographical prioritization of social network messages in near real-time using sensor data streams: an application to floods, *Computers and Geosciences* 80 (84-94).
- [6] A. M. Youssef, B. Pradhan, A. M. Hassan, Flash flood risk estimation along the st. katherine road, southern sinai, egypt using gis based morphometry and satellite imagery, *Environmental Earth Sciences* 62 (3) (2011) 611.
- [7] N. Robertson, T. Chan, Aerial image segmentation for flood risk analysis, in: 16th IEEE International Conference on Image Processing, 2009, pp. 597–600.
- [8] T. Hahmann, S. Martinis, A. Twele, A. Roth, M. Buchroithner, Extraction of water and flood areas from sar data, in: 7th European Conference on Synthetic Aperture Radar, 2008, pp. 1–4.
- [9] N. Tholey, S. Clandillon, P. D. Fraipont, The contribution of spaceborne sar and optical data in monitoring flood events: examples in northern and southern france, *Hydrological Processes* 11 (10) (1997) 1409–1413.
- [10] S. I. Khan, Y. Hong, J. J. Gourley, M. U. Khattak, T. D. Groeve, Multi-sensor imaging and space-ground cross-validation for 2010 flood along indus river, pakistan, *Remote Sensing* 6 (3) (2014) 2393–2407.
- [11] D. M. Tralli, R. G. Blom, V. Zlotnicki, A. Donnellan, D. L. Evans, Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards, *ISPRS Journal of Photogrammetry and Remote Sensing* 59 (4) (2005) 185–198.
- [12] C. van der Sande, S. de Jong, A. de Roo, A segmentation and classification approach of ikonos-2 imagery for land cover mapping to assist flood risk and flood damage assessment, *International Journal of Applied Earth Observation and Geoinformation* 4 (3) (2003) 217–229.
- [13] B. Ko, J. Kwak, J. H. Hong, J. Nam, Current trend in natural disaster warning systems based on computer vision techniques, in: Asian Conference on Pattern Recognition (ACPR), 2011.
- [14] [online, cited November 1, 2016][link].
- [15] H. K. Heidemann, Collection and Delineation of Spatial Data, U.S. Geological Survey, 2014, Ch. Lidar Base Specification.
- [16] S. Park, N. Lee, Y. Han, H. Hahn, The water level detection algorithm using the accumulated histogram with band pass filter, *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 3 (8) (2009) 2151–2155.

- [17] J. YU, H. HAHN, Remote detection and monitoring of a water level using narrow band channel, JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 26 (2010) 71.
- [18] R. Narayanan, V. M. Lekshmy, S. Rao, K. Sasidhar, A novel approach to urban flood monitoring using computer vision, in: 2014 International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2014.
- [19] M. A. Witherow, K. M. Iftekharuddin, M. Cetin, Analysis of crowd-sourced flooding images using computer vision techniques, Tech. rep., Old Dominion University USRIP Symposium (2016).
- [20] G. Baatzl, O. Saurer, K. Koser, M. Pollefeys, Large scale visual geo-localization of images in mountainous terrain, in: 12th European Conference on Computer Vision, 2012, pp. 517–530.
- [21] L. Baboud, M. Cadik, E. Eisemann, H. Seidel, Automatic photo-to-terrain alignment for the annotation of mountain pictures, in: Computer Vision and Pattern Recognition, 2011.
- [22] W. Liu, C. Su, Automatic peak recognition for mountain images, in: Advanced Technologies, Embedded and Multimedia for Human-centric Computing Advanced Technologies, Embedded and Multimedia for Human-centric Computing, Springer, 2014.
- [23] S. Ettinger, M. Nechyba, P. Ifju, M. Waszak, Vision-guided flight stability and control for micro air vehicles, in: Intelligent Robots and Systems, 2002.
- [24] P. Y. Shiznato, V. G. Jr, S. Osorio, D. F. Wolf, Fast visual road recognition and horizon detection using multiple artificial neural networks, in: Intelligent Vehicles Symposium, 2012.
- [25] S. Fefilatyevlatyev, V. Smarodzinava, L. O. Hall, D. B. Goldof, Horizon detection using machine learning techniques, in: 5th International Conference on Machine Learning and Applications, 2006.
- [26] B. Zafarifar, H. Weda, Horizon detection based on sky-color and edge features, in: SPIE, 2008.
- [27] R. O. Duda, P. E. Hart, Use of the hough transformation to detect lines and curves in pictures, Tech. rep., Artificial Intelligence Center at SRI International (1971).
- [28] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.
- [29] I. Sobel, G. Feldman, An isotropic 3x3 image gradient operator for image processing, presented at Stanford Artificial Intelligence Project (SAIL) (1968).
- [30] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, , S. Susstrunk, Slic superpixels compared to state-of-the-art superpixel methods, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, 2012.

- [31] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks* 61 (2015) 85–117.
- [32] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.
275
- [33] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587.
- [34] K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, *Journal of Big Data* 3 (9).
- [35] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, A. Yuille, The role of context for object detection and semantic segmentation in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
280