FLOOD-WATER LEVEL ESTIMATION FROM SOCIAL MEDIA IMAGES

P. Chaudhary[1], S. D'Aronco[1], M. Moy de Vitry[2], J. P. Leitão[2], J. D. Wegner[1]

[1] EcoVision Lab, Photogrammetry and Remote Sensing group, ETH Zürich, Switzerland
(priyanka.chaudhary, stefano.daronco, jan.wegner)@geod.baug.ethz.ch
[2] Department Urban Water Management, Eawag - Swiss Federal Institute of Aquatic Science and Technology, Switzerland
(matthew.moydevitry, joaopaulo.leitao)@eawag.ch

Commission II, WG II/6

KEY WORDS: Object detection, Deep learning, Image segmentation, Flood estimation, Instance segmentation, Flood detection

ABSTRACT:

In the event of a flood, being able to build accurate flood level maps is essential for supporting emergency plan operations. In order to build such maps, it is important to collect observations from the disaster area. Social media platforms can be useful sources of information in this case, as people located in the flood area tend to share text and pictures depicting the current situation. Developing an effective and fully automatized method able to retrieve data from social media and extract useful information in real-time is crucial for a quick and proper response to these catastrophic events. In this paper, we propose a method to quantify flood-water from images gathered from social media. If no prior information about the zone where the picture was taken is available, one possible way to estimate the flood level consists of assessing how much the objects appearing in the image are submerged in water. There are various factors that make this task difficult: *i*) the precise size of the objects appearing in the image might not be known; *ii*) flood-water appearing in different zones of the image scene might have different height; *iii*) objects may be only partially visible as they can be submerged in water. In order to solve these problems, we propose a method that first locates selected classes of objects whose sizes are approximately known, then, it leverages this property to estimate the water level. To prove the validity of this approach, we first build a flood-water image dataset, then we use it to train a deep learning model. We finally show the ability of our trained model to recognize objects and at the same time predict correctly flood-water level.

1. INTRODUCTION

Due to global climate change, flood events are predicted to become more frequent and damaging (Hirabayashi et al., 2013, Vitousek et al., 2017). In order 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 texts and pictures describing the situation. These data have the advantage to be cheap and available in real-time directly from the flooded region. The main disadvantage is that texts and pictures need to be properly processed in order to extract meaningful information. Although some studies, like for instance (Starkey et al., 2017, Aulov et al., 2014), already analyzed the problem of gathering flood information from social media, no methodology has been proposed yet to retrieve, in a fully automatic way, flood information from social media pictures. With this work, we aim at filling this gap, by proposing a deep learning framework to predict flood-water level from images.

Estimating flood level from images is not a trivial task. The main complication stems from the fact that the water level might not be univocal, as it can vary across different zones which are visible in the image. In order to work around this problem, we look separately at the different objects that appear in the picture and we estimate for each of them individually, how much they are submerged in water. If we then know an approximate height of the different objects, we can have a rough estimate of the water level. However, looking at objects individually, reveals other challenges: *i*) objects submerged in water might be only partially visible; *ii*) the size of the objects cannot be known accurately. It is important to be able to recognize objects even if they are only partially visible, as these objects retain the information about the water level. Fortunately, deep learning models such as the one we use, are still able to identify objects when the occlusion is not extremely large. In order to solve the second challenge, we look for some specific objects that are both common, so they are likely to appear in the pictures, and whose size is roughly constant among different instances. In this way, we can recover not only the water level relative to the object but also an absolute estimate.

The pipeline of the proposed method is the following. We first employ a state-of-the-art Convolutional Neural Network (CNN) architecture for object detection. Then, for the objects belonging to specific categories we estimate, again using a Neural Network architecture, how much they are submerged in water. After defining the network architecture we build a dataset by annotating images with flood-water level information and use it to train our network. We then evaluate the proposed model showing its ability to predict effectively water level for the objects in images. Finally, we further propose a method to merge the different water levels of the objects to obtain a single global level of the image.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W5-5-2019 | © Authors 2019. CC BY 4.0 License.

The rest of the paper is structured as follows. In Section 2, we review some related work. In Section 3, we illustrate in detail the proposed model. Whereas in Section 4 we describe in the annotation strategy used for the dataset and provide the experimental results. Finally, conclusions and future work are provided in Section 5.

2. RELATED WORK

2.1 Flood estimation from images and social media

As permanent surveillance systems for flood-water would incur large costs, recently several works investigated alternative options to retrieve flood information.

Starkley et al. (Starkey et al., 2017) demonstrate the importance of community-based observations also known as 'citizen science'. The observations used in this project were in many cases either photographs or videos. The work shows that community-based collected data is extremely valuable for local flash flood events. Unfortunately, in the proposed framework quantitative flood metrics are extracted manually. In our case we aim at developing a method that is able to retrieve information from images in a fully automatized way.

Fohringer et al. (Fohringer et al., 2015) propose a methodology for flood inundation mapping. They focus on images to extract inundation area and water depth information. The strength of this procedure is that information is readily available especially in urban areas. This is important as alternative information sources like analysis of remote sensing data, does not perform very well. The main weakness of the system is that, though being able to retrieve social media content automatically, it still requires to manually assess the relevance and plausibility of the content.

Other works such as Aulov et al. (Aulov et al., 2014) and ZhenLong et al. (Li et al., 2018) also suggest to gather information from social media platforms in order to obtain useful information and be able to build a flood map. The method proposed by Aulov et al., in particular, is able to determine the regions which were flooded, together with a rough estimate of the surge levels, as well as the regions free from flood by manually inspecting street photos. ZhenLong et al. (Li et al., 2018) instead use georeferenced social media texts to extract information and combine it with a digital elevation model to generate a flood map

Probably the closest work to ours is the one of Kröhnert et al. (Kröhnert and Eltner, 2018). They propose to use of smartphones and other fixed embedded system cameras to estimate water level. The proposed method is able to achieve extremely good results and reach accuracy levels for water stage measurements in the order of millimeters. The main limitation of this approach is that it requires the digital surface model of the scene to be available in order to make estimates based on pictures taken using a smartphone. In our work, instead, we drop this assumption and do not consider any prior knowledge of the scene.

2.2 CNN for Object Detection

In the last years deep learning has established as the state-of-the-art tool for several image-centric tasks. In this work we are mainly interested in object detection and classification. One of the most successful methods for object detection is the Region-based CNN (R-CNN) approach, proposed in (Girshick et al., 2013). R-CNN first generates a number of candidates object regions and then uses a neural network to classify independently the objects appearing in such regions. As further improvement, Fast and Faster R-CNN (Girshick, 2015, Ren et al., 2015) are modifications of the R-CNN architecture which use shared features, among the region proposal and the final classification, in order to make the network faster and more efficient. Finally, Mask R-CNN (He et al., 2017) extends the Faster R-CNN algorithm to further provide a segmentation mask for the different objects detected in the image. Mask R-CNN represents the basic architecture which we extend in order to obtain the flood-water level prediction. Further details on the Mask R-CNN architecture are provided in the next section.

3. METHODOLOGY

In this section, we describe the deep learning approach used for flood-water level estimation. We use Mask R-CNN (He et al., 2017) as base architecture which is a state-of-the-art solution for instance segmentation. Figure 1 illustrates the overall architecture of the method. The backbone of the architecture works as the main feature extractor. We can use any standard convolutional neural network. The idea is to pass an image through various layers which extract different features from the image. The lower layers detect low-level features like blobs, edges. As we move to higher layers, they start detecting full objects like cars, people, buses. The input image gets converted to feature maps in this module for an easier handling in the other modules (Abdulla, 2018). The above described backbone can be improved upon using Feature Pyramid Network(FPN) (Lin et al., 2016) which was introduced by the same authors of Mask R-CNN (He et al., 2017). FPN represents objects at multiple scales better by passing the high level features from first pyramid down to lower layers of second pyramid. This allows features to have access to both lower and higher level features. We use ResNet101 (He et al., 2015) and FPN as our backbone (Abdulla, 2018).

The Region proposal network (RPN) is a neural network which scans over the image and gives scores based on whether there is an object or not in the scanned regions. The regions are known as *anchors*, and they are basically boxes covering the image. There are thousands of anchors enveloping the image of different sizes and aspect ratios. These anchors are classified as positive, neutral or negative based on their scores. Anchors with high score (positive anchors) are then sent to the next stage for classification. RPN also scans over the feature maps generated by the backbone instead of the image to avoid multiple calculations. It might be possible that positive anchors do not cover an object completely. To resolve that, RPN does refinement on the anchors (Abdulla, 2018). As the anchors are sometimes lying very close to each other, and they have a high degree of overlap, we suppress some bounding boxes per class for better results. To perform this task we apply non-maximal suppression technique. This technique first calculates the intersection over union of these anchors, then, if the value is higher than a certain threshold and if the boxes belong to the same class, we keep only the one with higher object score.The output of this first network stage are a set of Regions of Interest (RoIs) which are fed to the next stage for proposal classification.

In our adaptation of the Mask R-CNN, the proposal classification generates overall three outputs for each RoI:

- **Class**: In contrast to RPN, where we try to know whether

https://doi.org/10.5194/isprs-annals-IV-2-W5-5-2019 | © Authors 2019. CC BY 4.0 License.

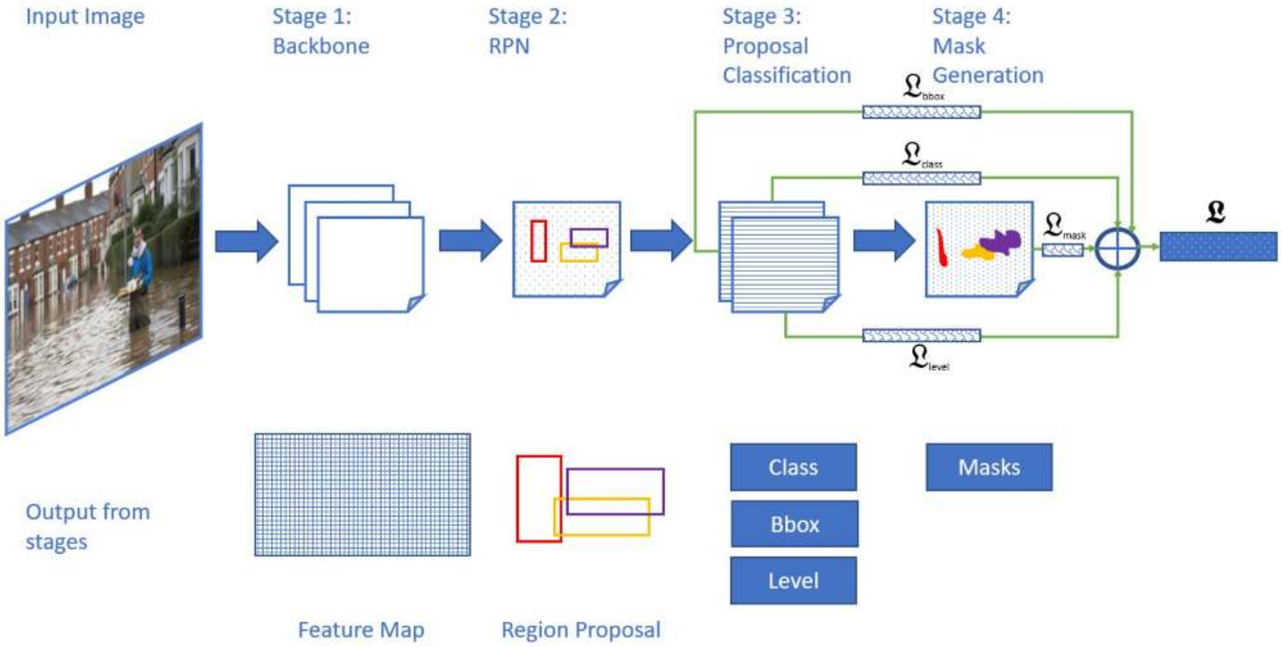


Figure 1. Shows the overall architecture and respective output from each stages

there is an object or not, here we classify the particular object. If the class turns out to be background, we drop the proposal.

- **Bounding Box Regression**: We further try to refine the bounding box for each classified proposal to obtain a more accurate position.
- **Flood Level**: We predict the flood level class of the proposal.

As additional output, Mask R-CNN also generates a binary mask to perform object segmentation on the object contained inside each RoI. The mask branch is a small convolutional network which is applied to each RoI and works in parallel to the RoI class, water level, and bounding box offset predictions. The total loss is the sum of individual losses and it is written mathematically as:

$$\mathcal{L} = \mathcal{L}_{\text{class}} + \mathcal{L}_{\text{bbox}} + \mathcal{L}_{\text{level}} + \mathcal{L}_{\text{mask}} \tag{1}$$

where, $\mathcal{L}_{\text{class}}$, $\mathcal{L}_{\text{bbox}}$, $\mathcal{L}_{\text{mask}}$ are defined as in (He et al., 2017) and (Ren et al., 2015). $\mathcal{L}_{\text{mask}}$ is defined as the average binary cross-entropy loss for each associated RoI with ground-truth class k. $\mathcal{L}_{\text{mask}}$ is only defined for the k^{th} mask, which means other masks generated do not contribute to the mask loss. This definition of $\mathcal{L}_{\text{mask}}$ allows the network to generate masks for every class without competition among classes. We use cross entropy as loss function for the level prediction:

$$\mathcal{L}_{\text{level}} = \frac{1}{N_{\text{level}}} \sum_{i}^{N_i} [- \sum_{l}^{L} x_{il} \log{(q_{il})}] \tag{2}$$

where, N_i is the number of RoIs of the image, x_{il} is a binary variable equal to one if anchor i belongs to level class l, q_{il} is the predicted score for anchor i belonging to class l, and L is the total number of level classes. For the level branch, we use two fully connected layers interlaced with batch normalization layers and non-linear layers, followed by a softmax activation layer whose output represents us the probability that any of the classes are true.

4. EXPERIMENTS

In this section we first introduce the annotation strategy and the dataset used for training the network. We then show some experimental results to evaluate the performance of the proposed method.

4.1 Annotation strategy

In order to train our neural network we need the images in our training dataset to be labeled for all the four quantities we want to predict.

As the goal of this study is to quantify flood-water level based on objects partially submerged in water, the first step for defining the annotation strategy is to decide which objects we should consider for the classification task. The criteria for selecting the objects for this task are: easy availability, known dimensions, and low intra-class height variance. By easy availability, we mean objects which are common in the real world, so that it is easy to gather a large number of pictures containing the object, both for training and prediction. Known dimensions refer to the fact that height, length and width of an object are approximately known. Finally low variation in height means that several instances of the same object in the real world have approximately the same height. For instance, bicycles are objects that are extremely common in urban environments, we roughly know their size, and their height is roughly constant across different models. Based on the criteria we decided to consider these five classes of objects: **Person**, **Car**, **Bus**, **Bicycle**, and **House**. In addition to the five classes mentioned above, we also consider the **flood** class, which represents flood-water present in the image. The insight behind adding this class is that the feature extraction stage of the neural network might leverage this information to create better feature maps, which ultimately lead to a better flood-water level prediction. For each of these objects appearing in the dataset images we further define a bounding box containing the object and a segmentation mask which highlights the object.

The one label we are missing is the one for the water level prediction. To obtain this label we need a course of action to quantify the flood-water height. As humans cannot just by looking at an image deduce the centimetres of flood-water, we decided to pursue a strategy that tries to estimate how much of an object body is submerged in water in terms of some coarsely defined levels. Since the main concern in case of floods is to prevent human fatalities, it makes sense to consider the human size as main building block for this prediction. We consider 11 flood levels, levels go from 0, which means no water, to 10, which represents a human body of average height completely submerged in water. Moreover, since in order to create the training dataset, we need to annotate manually the images with water level information, it is important to select levels that facilitate this operation. The height of the different levels is then inspired by drawing artists who use head height as the building block for the human figure. To map level classes to actual flood height, we consider an average height human body and derive the water height in cm, see Table 1. We can now extend the annotation strategy to the other four different classes of objects by considering their average height. After getting an approximate estimate of the height of these objects in the real world, we compare them with the average human height, on which the 11 flood levels are defined, and extend the flood level definition to these other objects. In Figure 2 we show how, the flood levels defined for a human body, translate for an average size Bicycle.

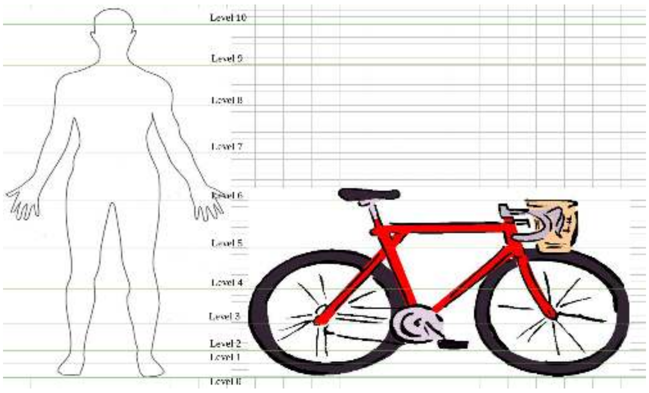


Figure 2. The figure shows the annotation strategy for object Person and Bicycle

4.2 Datasets

We make use of two datasets in this work. The first dataset contains images of flood events which from henceforth will be referred to as Flood Dataset, the second one is the MS COCO Dataset (Lin et al., 2014). In order to feed the annotated datasets to the network, we use MS COCO object detection annotation format (COCO, n.d.). We extract the information for various fields from the annotated image and generate mask for each object instance using MS COCO API (cocoapi, n.d.). The annotations are stored using JSON (JSON, n.d.).

4.2.1 Flood dataset For the ground truth generation, the Flood dataset is pixel-wise annotated using an online annotation tool called Supervisely (Supervisely, n.d.). Pixel-wise annotation means that every pixel of the image is assigned to a specific instance of a class. The images of the Flood dataset have at least one of the five objects considered for the level estimation in them. We also did not consider any grayscale or aerial view image. The Flood dataset is composed of 7000 unique images. The images have been manually collected from various different sources such as Google, Flickr and National Geographic.

Level Name	Range	Value nearest integer
	cm	cm
level0	No water	0.0
level1	0.0 - 1.0	1.0
level2	1.0 - 10.0	10.0
level3	10.0 - 21.25	21.0
level4	21.25 - 42.5	43.0
level5	42.5 - 63.75	64.0
level6	63.75 - 85	85.0
level7	85.0 - 106.25	106.0
level8	106.25 - 127.5	128.0
level9	127.5 - 148.75	149.0
level10	148.75 - 170.0	170.0

Table 1. Level to centimeters relation table

4.2.2 MS COCO dataset MS COCO (Lin et al., 2014) is a large-scale dataset for object detection, segmentation and captioning. It was released in 2015 and prepared by Lin et al. (Lin et al., 2014). For our task, we use the year 2017 version (COCO, 2018) which has 118 thousand train images and 5 thousand validation images (COCO, 2018). It has only four of our selected object categories(**Person**, **Car**, **Bus**, and **Bicycle**) and no **House** category. So from this dataset we consider, for training and validation, only images containing at least one instance of these categories.

4.3 Evaluation strategy

As our network generates four different predictions (class, bounding box, level, and mask prediction) but ultimately we only care about the level prediction, we need to separate the performance of the object detector and level classifier. There can be two scenarios: False Positive(FP) and False Negative(FN) detections. If an object instance is not detected, there will also be no level predictions. In simple terms, the ground-truth file stores for every image, the bounding box, class label, level label, and mask, for all object instances present in the image. During prediction, if one or more of these object instance(s) is not detected there will also be no level label prediction. This scenario corresponds to the False Negative(FN) case. Similarly, False Positive cases are also a possibility, in this case an object detector wrongly detects background (image area where no object lies) as an object and predicts a class label for background. If a class of an object is wrongly predicted, it is also considered a FP case. In Figure 6, we show example of both the FP and FN case.

We describe now a method to compute a global image water level from the individual object predictions. The reason for doing this is that for our purpose, we ultimately do not need a level value for each object instance, as it is too detailed, but an overall score for the entire image might be sufficient. The question that obviously arises is why not just train the network to give a single value for every image. At first we investigated this naive approach which, however, performed poorly. It is in fact very complicated to predict directly a water level for the entire image. For example, in

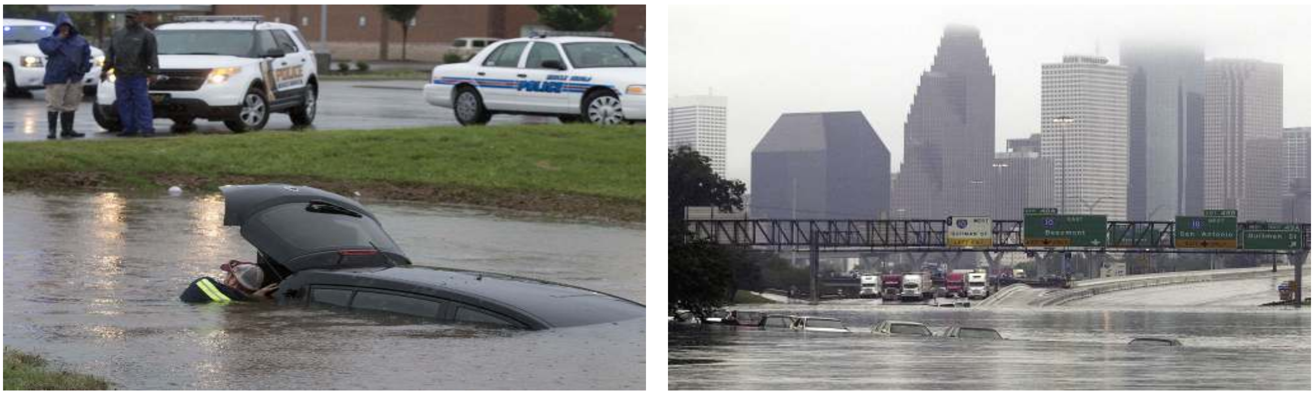


Figure 3. Shows two images from flood dataset with varying flood height in the image

Figure 3 we can see that flood height is not consistent throughout the image. If we just predict a single output value we cannot tell why the model predicted that. To improve system performance we therefore switched to an approach that predicts per object water level and then combines multiple level predictions into a global one.

For the calculation of the global water level we compute the trimmed mean of the predicted levels of the different object instances. The Trimmed mean or truncated mean is defined mathematically as:

$$T = \frac{1}{n * (1 - 2\alpha)} \left((1 - r) * (X_{g+1} + X_{n-g}) + \sum_{i=g+2}^{n-g-1} X_i \right), \tag{3}$$

where X is a sorted numerical vector (which in our case represents the level of the different object instances), n is the length of X, α is the proportion of trim from each end, g is the integer part of the $n * \alpha$ and r is fractional part of $n * \alpha$. The reason for using trimmed mean over mean and median is that mean is very sensitive to outliers.

Taking all the above mentioned reasons into consideration, we performed three experiments. The reason for performing these operations is to decouple the class prediction and level prediction performance. The experiments are described in detail below:

- **Experiment 1**: In this experiment, from each image's prediction and ground-truth level values, we remove the FP and FN cases, which we have described above. The FP cases are removed from the level prediction based on its definition, and the FN cases are removed from ground-truth values. For the remaining object instances, we take separate trimmed mean to recover a single value for the prediction and ground-truth level. We convert the trimmed mean values to centimeters (cms) using Table 1. Then we take the absolute difference between the predicted level height (in cms) and ground-truth level height (in cms). We do this for all images in the test dataset and take the mean of all the absolute differences generated for the test dataset. This gives the mean error in cms of the flood height for a single image.

- **Experiment 2**: For Experiment 2, we repeat the same steps as Experiment 1, but in this case we keep the FP and FN cases. This means we do not do any alterations to ground-truth and predicted level values. We take the trimmed mean of predicted level values and ground-truth values separately. Then follow the same procedure as described in Experiment 1.

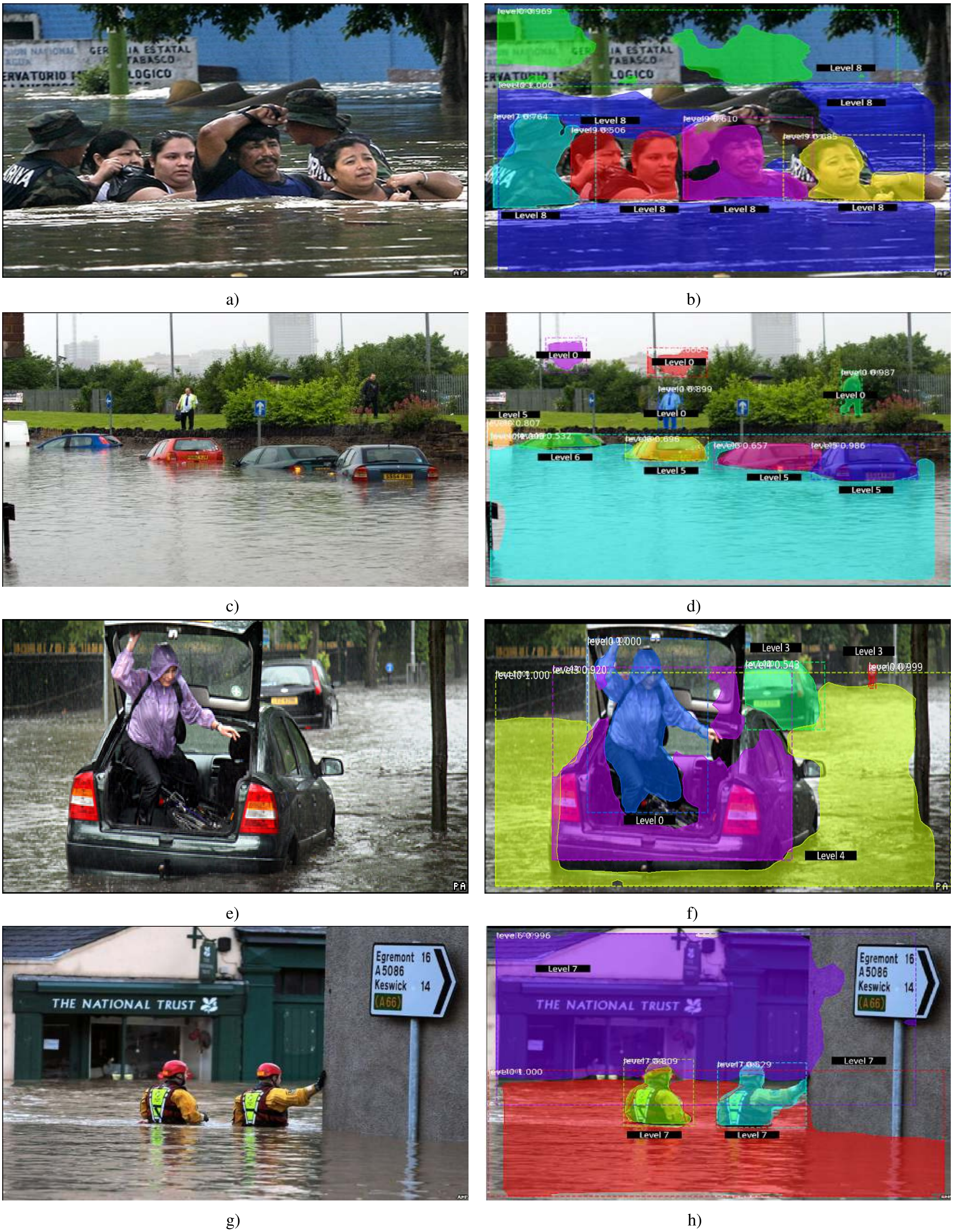
- **Experiment 3**: In the third experiment also, we do not remove any entries from prediction or ground-truth level values, but for every class prediction entry which has no match in ground-truth, i.e., FP case, we try to find a reference for level prediction. When a background, or object, is wrongly classified as some other class, we have no respective entry in the ground-truth level. So to calculate how well the level predictor performs even if the class prediction is wrong we find the nearest object instance in the ground-truth and use its ground-truth level as true value for the wrongly predicted class. To do this we calculate the distance between the wrongly classified object bounding box and all the other object instances with a level prediction in the image, we then take the level label from the nearest labeled object.

4.4 Results

To build the training dataset we keep the ratio of images from the Flood dataset and the images from the MS COCO dataset balanced. We train the model using this dataset (standard), We further perform k-fold cross validation on the training dataset. In order to limit the amount of computations for training and, at the same time, reduce the bias we use a 5-fold cross validation.

The Table 2 shows the summary of the experiments performed. The values for different experiments are calculated using the procedure described in Section 4.3. The **Exp 1**, **Exp 2**, and **Exp 3** column names refer to the three experiments previously defined. We can see that Experiment 1, which corresponds to the case where we discard all FP and FN cases for calculating level value per image, has the lowest error values. For Experiments 2 and 3 the average cross validation model performs better than the standard one. It is also worth noting that for cross validation model Experiment 3 and Experiment 1 error values are much closer (difference 0.24 cm) than the standard model's difference of 2.15 cm. More importantly, for all the experiments, the mean absolute error evaluated on the test dateset is always smaller than 10 cm, which can be considered an acceptable level of accuracy for this task.

Figure 4 and Figure 5 show qualitative evaluations for some of the test images. In the images we show the ground-truth level label in the black boxes for easier evaluation and comparison. Also, note that, the mask colors in the following figures carry



a)

b)

c)

d)

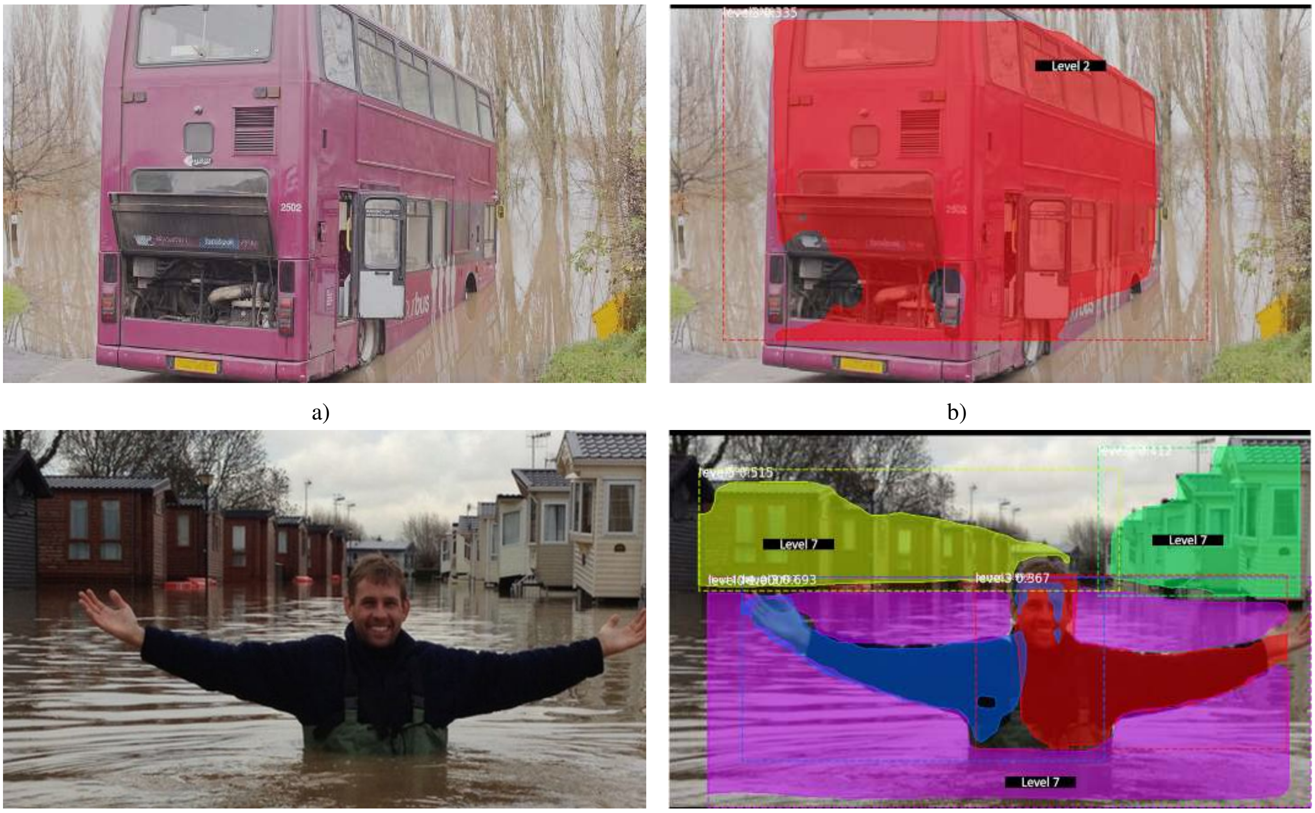
e)

f)

g)

h)

Figure 4. Shows qualitative evaluation of test images.



a) b)

c) d)

Figure 5. Contd. qualitative evaluation of test images.

		Exp 1	Exp 2	Exp 3
Error(cm) in standard model		7.32	9.47	9.47
Error(cm) in 5-fold cross validation(CV)	fold1	8.80	9.76	8.79
	fold2	7.25	7.40	6.82
	fold3	6.88	7.86	8.29
	fold4	7.91	9.60	8.70
	fold5	8.32	8.16	7.76
Mean error(cm) in 5-fold cross validation		**7.83**	**8.56**	**8.07**

Table 2. Summary of experiments

no particular meaning. It is generally observed that in images of flood events, objects are likely to be partially occluded and cluttered. So, it is important for the model to perform well in such cases. In Figure 4*(b)* we show one such image. We can observe that in 4*(b)*, the objects are highly occluded as only small portion of their bodies are visible and they are standing very close together. This makes the detection task harder. As we can see from the prediction, in this case, two persons are detected as a single one, and another person is not detected at all. Though this is not an ideal detection result, for our purpose it is not necessary to detect each and every one of the object instances in the image, as we mainly care about predicting the correct water level.

It is also common to see during flood events, people standing or sitting on objects or on elevated surfaces. The low lying areas are flooded first and people often try to reach the elevated areas to protect themselves. The flood-water level is not necessarily univocal in the entire image. The figure 4*(d)* shows exactly such a scenario and it is important to identify and correctly detect such cases in the image. As not all objects in a flood event image are partially submerged, and accurate prediction of those cases enhances the performance of the model greatly. In figure 4*(d)*, we can see two persons on higher grounds that are correctly classified, whereas some of the cars are classified as level 6 instead of level 5 and vice versa.

Similarly in Figure 4*(f)*, we see a person standing in the back of the car and it has been correctly predicted as level 0. The cars predictions though are not fully accurate. It is predicted level 4 when the ground-truth is level 3 and vice versa for the other car. In Figure 4*(h)*, we see an image from a flood event where there are three objects and except one object being wrongly predicted as level 6 instead of level 7, other objects are correctly predicted.

In Figure 5, we show two examples of poor performance of the model. In Figure 5*(b)*, the bus is wrongly classified as a house which can be due to presence of large windows and a door. Also, there is no class flood detected even though we can see floodwater in the image. This might be because the color of floodwater is brown and also due to the stillness of water, which are not usual features for a water body. In the second image, Figure 5*(d)*, we see a single person classified as two persons as well as a wrong level prediction.

5. CONCLUSIONS

In this paper, we have presented a model to predict flood-water level from images gathered from social media platforms in a fully automatized way. The prediction is done using a deep learning framework. More specifically we have build this model on top of the Mask R-CNN architecture. The proposed model performs instance segmentation and at the same time predicts flood level whenever an instance of some specific objects is detected. We

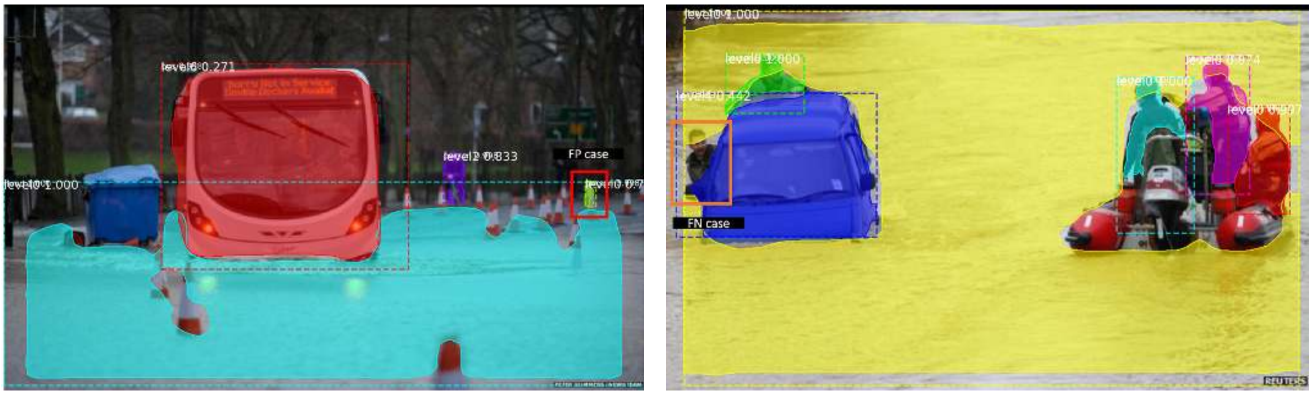


Figure 6. Shows two image prediction results. *Left*: FP case, where a road barrier is predicted as a person shown in red box. *Right*: FN case, where one person is not detected shown using orange box.

further provide a method to combine the multiple object instances level predictions and obtain a single water level prediction for the entire image. The conducted experiments proved the ability of the trained model to effectively predict water level from images within an acceptable error.

As future work we plan to extend our framework in order to leverage also text information. Indeed, images posted on social media platforms are often associated with some text related to the picture. The insight is that if we could combine these two related information, we would be able to further improve the prediction accuracy. On a different path, we also plan to investigate more advanced methods about how to combine the water level predictions for each object instance, in order to obtain a global image level.

REFERENCES

Abdulla, W., Splash of color: Instance segmentation with mask r-cnn and tensorflow. https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46 (Last visited on: 16/01/2019).

Aulov, O., Price, A. and Halem, M., 2014. Asonmaps: A platform for aggregation visualization and analysis of disaster related human sensor network observations. In: *ISCRAM*.

COCO, M., Common objects in context(coco). http://cocodataset.org/#home (Last visited on: 16/01/2019).

COCO, M., Ms coco annotation format. http://cocodataset.org/#format-data (Last visited on: 16/01/2019).

cocoapi, cocoapi. https://github.com/cocodataset/cocoapi (Last visited on: 16/01/2019).

Fohringer, J., Dransch, D., Kreibich, H. and Schröter, K., 2015. Social media as an information source for rapid flood inundation mapping. *Natural Hazards and Earth System Sciences* 15(12), pp. 2725–2738.

Girshick, R., 2015. Fast R-CNN. In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Girshick, R. B., Donahue, J., Darrell, T. and Malik, J., 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*.

He, K., Gkioxari, G., Dollár, P. and Girshick, R. B., 2017. Mask R-CNN. *CoRR*.

He, K., Zhang, X., Ren, S. and Sun, J., 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

Hirabayashi, Y., Roobavannan, M., Koirala, S., Konoshima, L., Yamazaki, D., Watanabe, S., Kim, H. and Kanae, S., 2013. Global flood risk under climate change. 3, pp. 816–821.

JSON, Json. http://json.org/ (Last visited on: 16/01/2019).

Kröhnert, M. and Eltner, A., 2018. Versatile mobile and stationary low-cost approaches for hydrological measurements. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2, pp. 543–550.

Li, Z., Wang, C., Emrich, C. T. and Guo, D., 2018. A novel approach to leveraging social media for rapid flood mapping: a case study of the 2015 south carolina floods. *Cartography and Geographic Information Science* 45(2), pp. 97–110.

Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B. and Belongie, S. J., 2016. Feature pyramid networks for object detection. *CoRR*.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L., 2014. Microsoft COCO: common objects in context. *CoRR*.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In: *Neural Information Processing Systems (NIPS)*.

Starkey, E., Parkin, G., Birkinshaw, S., Large, A., Quinn, P. and Gibson, C., 2017. Demonstrating the value of community-based (citizen science) observations for catchment modelling and characterisation. *Journal of Hydrology* 548, pp. 801 – 817.

Supervisely, Supervisely annotation tool. https://supervise.ly (Last visited on: 16/01/2019).

Vitousek, S., Barnard, P. L., Fletcher, C. H., Frazer, N., Erikson, L. and Storlazzi, C. D., 2017. Doubling of coastal flooding frequency within decades due to sea-level rise. *Scientific reports* 7(1), pp. 1399.