Flood detection from social multimedia and satellite images using ensemble and transfer learning with CNN architectures

Danielle Dias, Ulisses Dias
University of Campinas, Brazil
danielle.dias@ic.unicamp.br,ulisses@ft.unicamp.br

ABSTRACT

In this paper we explore deep convolutional neural networks pre-trained on ImageNet along with transfer learning mechanism to detect if an area has been affected by a flood in terms of access. We worked in two tasks with different datasets. The first dataset contains images from social media and the goal is to identify direct evidence for passability of roads by conventional means. The second dataset contains high resolution satellite imagery of partially flooded areas and the goal is to identify sections of roads that are potentially blocked. For both tasks, we used visual information only and our best models achieved averaged F1-Score value of 64.81% on the first task and 73.27% on the second task.

1 INTRODUCTION

Flooding events demand fast response. Rescue and medical teams should move fast to the affected points and bring victims to safety in a timely manner. Unfortunately, roads may be affected by the flood in terms of access. Automatic road passability recognition aids the support planning that will mitigate the impact of disasters.

The "Multimedia Satellite Task 2018" studies the problem of road passability classification, namely whether or not it is possible to travel through a flooded region. Two tasks were proposed depending on the source of information. In the first task, we should take advantage of the high popularity of social media and filter those information which provide direct evidence for passability of roads. In the second task, we receive high resolution satellite imagery of partially flooded areas and the goal is to identify if it is possible to go from a point A to a point B. More details can be found in the task overview [1].

2 APPROACH

The dataset for the social media task consists of 7,387 images and the dataset for the remote sensing task consists of 1,664 satellite images. As the size of the dataset is limited, we decided to use the transfer learning mechanism in both cases in a similar workflow: images are received as input, pre-trained convolutional neural networks (CNNs) are used for feature extraction (Section 2.1), artificial neural networks (ANNs) predict labels (Section 2.2), and an ensemble is constructed of individual classifiers (Section 2.3).

While in the social media subtask images are the only source of information, in the remote sensing subtask we receive the images along with two points A and B. The question is whether or not we can go from point A to point B. Thus, we preprocess the images to embed these points within the image (Section 2.4).

MediaEval'18, 29-31 October 2018, Sophia Antipolis, France

2.1 Transfer Learning Mechanism

Many advanced CNN architectures have been trained on ImageNet and are currently available. We selected 10 of them as feature extractors: DenseNet121 [5], DenseNet169 [5], DenseNet201 [5], InceptionResNetV2 [8], InceptionV3 [9], MobileNet [3], ResNet50 [4], VGG16 [7], VGG19 [7], Xception [2]. We also studied if global feature based approaches extracted with Lire [6] could provide any significant advantage to the pre-trained models, but since no improvement was achieved, we decided to use only features extracted from CNNs.

We replaced the architecture prediction layers with new ANN models, which are responsible for returning the classification labels. For the social media subtask, the output labels account for (i) no evidence, (ii) evidence/not passable, and (iii) evidence/passable. For the remote sensing subtask, the output labels account for (i) passable, and (ii) non passable. That said, ANNs in the former subtask have three units, whereas ANNs in the latter have only two. The output layers use softmax activation function.

2.2 Prediction Layer Models

Two approaches performed best on our 5-fold cross validation analysis of prediction layers. They are hereby called Model_1 and Model_2.

Model_1 is an ANN having only one hidden layer with 512 nodes. Each node uses ReLU as activation function. We added a Dropout layer with a dropout ratio of 50% in the hidden layer, and l_2 regularization to prevent overfitting.

Model_2 is an ANN having two hidden layers. The first has 2048 nodes and the second has 128. Nodes in hidden layers use ReLU as activation function, and l_2 as regularization. We dropped out 80% of the connections between input layer and the first hidden layer. We also added a dropout ratio of 50% in each hidden layer.

2.3 Ensemble

We have 10 CNN architectures to extract features and 2 ANN architectures for prediction. Therefore, for each image we have 20 class predictions, each prediction is a vector of three floating point numbers in the social media task and two floating point numbers in the remote sensing task.

To create an ensemble, we concatenate the class predictions and use logistic regression to map the $20 \times 3 = 60$ dimension vector to 3 output classes in the social media task, and to map the $20 \times 2 = 40$ dimension vector to 2 output classes in the remote sensing task.

2.4 Preprocessing Satellite Images

In Figure 1 we illustrate all the steps to preprocess the satellite images. In Figure 1(a) we show one of the images in the development dataset. Figures 1(b-d) have marks added for illustrative purposes

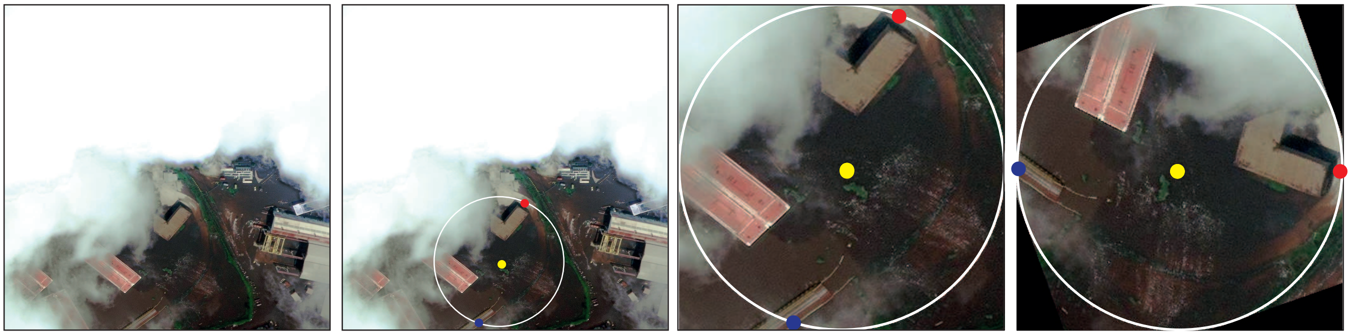

Figure 1: Preprocessing steps using the points A and B. In (a) we show an original image as given by the organizers. In (b) we add blue, red, and yellow marks to represent the points A, B, and the midpoint between A and B, respectively. We also draw a circunference centered in the midpoint that has diameter equal to the distance between A and B. In (c) we crop the image to include the entire circunference. In (d) we rotate the image to have A in the left side and B in the right side.

and to clarify the explanation, we did not really add these marks to the image provided to the CNNs. Blue and red marks represent the inputted A and B points, respectively. Our ultimate goals is to place these points in fixed locations, so the model could learn how to find a path between them. We follow by describing each step.

(1) We first compute a point C which is halfway between A and B as shown by the yellow mark in Figure 1(b).
(2) We compute a circunference centered in C that have the distance between A and B as diameter. Observe in Figure 1(b) that only a small area of the image is inside the circunference and that the area outside is not helpful to answer whether there is a path between A and B. This observation occurs in several cases.
(3) We crop the image to keep only the circunscript square as shown in Figure 1(c).
(4) We rotate the image around C to place A in the left side where the circunference touches the circunscript square, and B in the right side counterpart, as shown in Figure 1(d).

3 RESULTS AND ANALYSIS

During the training phase, we evaluated the models using 5-fold cross validation. We selected the four best models and the ensemble to submit to the organizers, which performed their analysis on unseen data and reported the results back to us. Table 1 and Table 2 present the results using the averaged F1-Score metric for the social media and remote sensing subtasks, respectively.

In the social media task, the ensemble produced the best results, obtaining F1-Score of 64.81% against 62.93% yielded by ResNet50, the best individual model. In the remote sensing task, the ensemble achieved 71.72% while the best individual model DenseNet121 reached 73.27%. We believe there is room for improvement if we tune the ensemble again, or if we replace the logistic regressor by other classification methods.

4 CONCLUSION

We used pre-trained CNNs as a starting point to create models that predict if it is possible to travel through a flooded area.

Table 1: Evaluation results for the flood classification task from social multimedia images. We highlight in bold the best result, which was achieved by the ensemble.

CNN Arch.	ANN Arch.	Averaged F1-Score (%)
DenseNet201	$Model_1$	62.82
VGG19	$Model_1$	60.92
Resnet50	$Model_1$	62.93
DenseNet169	$Model_1$	62.91
Ensemble	-	**64.81**

Table 2: Evaluation results for the flood classification task from satellite imagery. We highlight in bold the best result, which was achieved by DenseNet121 with $Model_1$.

CNN Arch.	ANN Arch.	Averaged F1-Score (%)
MobileNet	$Model_1$	56.82
MobileNet	$Model_2$	68.63
InceptionV3	$Model_2$	62.69
DenseNet121	$Model_1$	**73.27**
Ensemble	-	71.72

We combined features extracted from 10 CNNs with 2 models based on ANNs for prediction, then we build an ensemble by concatenating the predicted classes and using logistic regression to map them to a new output. This ensemble achieved best results in the social media task, but not in the remote sensing. Our results support the idea that transfer learning mechanism and ensemble are promising approaches for both tasks.

ACKNOWLEDGMENTS

We thank CAPES and CNPq (grant 400487/2016-0) and FAPESP (grant 2015/11937-9).

REFERENCES

[1] Benjamin Bischke, Patrick Helber, Zhengyu Zhao, Jens de Bruijn, and Damian Borth. The Multimedia Satellite Task at MediaEval 2018: Emergency Response for Flooding Events. In *Proc. of the MediaEval 2018 Workshop* (Oct. 29-31, 2018). Sophia-Antipolis, France.

[2] François Chollet. 2016. Xception: Deep Learning with Depthwise Separable Convolutions. *CoRR* abs/1610.02357 (2016). arXiv:1610.02357 http://arxiv.org/abs/1610.02357

[3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. (04 2017).

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[5] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. 2016. Densely Connected Convolutional Networks. *CoRR* abs/1608.06993 (2016). arXiv:1608.06993 http://arxiv.org/abs/1608.06993

[6] Mathias Lux and Savvas A. Chatzichristofis. 2008. Lire: Lucene Image Retrieval: An Extensible Java CBIR Library. In *Proceedings of the 16th ACM International Conference on Multimedia (MM '08)*. ACM, New York, NY, USA, 1085–1088.

[7] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014). arXiv:1409.1556 http://arxiv.org/abs/1409.1556

[8] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *CoRR* abs/1602.07261 (2016). arXiv:1602.07261 http://arxiv.org/abs/1602.07261

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. *CoRR* abs/1512.00567 (2015). arXiv:1512.00567 http://arxiv.org/abs/1512.00567