Data-Driven Flood Detection using Neural Networks

Keiller Nogueira[1], Samuel G. Fadel[2], Ícaro C. Dourado[2], Rafael de O. Werneck[2],
Javier A. V. Muñoz[2], Otávio A. B. Penatti[3], Rodrigo T. Calumby[2,4], Lin Tzy Li[2,3],
Jefersson A. dos Santos[1], Ricardo da S. Torres[2]
[1]Universidade Federal de Minas Gerais (UFMG), [2]University of Campinas (Unicamp),
[3]SAMSUNG R&D Institute Brazil, [4]University of Feira de Santana
[keiller.nogueira,jefersson]@dcc.ufmg.br,[samuel.fadel,icaro.dourado,rafael.werneck,lintzyli,rtorres]@ic.unicamp.br,
jalvarm.acm@gmail.com,o.penatti@samsung.com,rtcalumby@ecomp.uefs.br

ABSTRACT

This paper describes the approaches used by our team (MultiBrasil) for the Multimedia Satellite Task at MediaEval 2017. For both disaster image retrieval and flood-detection in satellite images, we employ neural networks for end-to-end learning. Specifically, for the first subtask, we exploit Convolutional Networks and Relation Networks while, for the latter, dilated Convolutional Networks were employed.

1 INTRODUCTION

Natural disaster monitoring is a fundamental task to create prevention strategies, as well as to help authorities to act in the control of damages. In its first appearance at MediaEval, the Multimedia Satellite Task [3] focuses on monitoring of flooding events, which is considered the most harmful and costly type of natural disaster in the world [8].The task is subdivided into two subtasks: (a) Disaster Image Retrieval from Social Media (**DIRSM**), which deals with flooding events in data (visual and textual) crawled from social media; and (b) Flooding-Detection in Satellite Images (**FDSI**), which refers to segment flooding regions in satellites images.

2 DISASTER IMAGE RETRIEVAL (DIRSM)

For the **DIRSM** task, we employed Convolutional Networks (CNN) [6] to deal with visual features. For textual features, we applied Relation Networks (RN) [9] and traditional methods (as baseline) such as Bag of Words (BoW) and bigrams. The recently proposed RN is a neural network designed for taking into account the relationship between pairs of objects during training. A RN consists of two neural networks, f and g, whose parameters are learned jointly.

In runs 1, 2, 3, and 4, we used neural networks and trained them for classification, with the positive class being a flooding event. In those runs, the final ranking was created by sorting the test set with respect to the classification score, from highest to lowest. Thus, ideally, images with flooding events should have higher score and appear first in the ranking. None of the runs used additional datasets.

Run 1: This run, which focuses only on visual data, employed GoogleNet [11] pre-trained on ImageNet dataset. We fine-tuned the network using the whole training set, replacing the original last layer by a new one containing two neurons, which correspond to the two classes: “flooding” and “non-flooding”.

Run 2: A relevant portion of the available metadata are tags and descriptions that are not necessarily well-written sentences. In addition, the amount of available data also discourages the use of recent neural networks designed for learning from text data, as many of them are large architectures based on convolutional or recurrent neural networks, thus requiring larger datasets of structured sentences. With this in mind, we hypothesized that the co-occurrence of words is still valuable evidence of a flooding event and is easier to learn from than structured sentences. For runs 2 and 3, which use text data, we extracted a set of words from each image metadata and used a neural network to learn from that set how to classify if an image describes or not a flooding event. This set is the union of the set of words extracted from the description and the set of words occurring in tags. To obtain the set of words from the description, we remove any HTML, non-letter symbols, stop words, and then apply a standard stemming procedure [2].

We designed a RN to learn from the sets of words for run 2. In order to create a representation for words, we built a word dictionary from all words in the training data, assigning an integer ID to each word. The first layer of the network is a fully connected layer that takes one-hot encoded vectors representing words as input (ID of the word is the index of the ‘1’ value) and outputs a 32-dimensional vector. Using this strategy, vector representations improve as learning goes on, while not requiring large word datasets.

Run 3: Since we had access to both images and their metadata, we use an architecture that incorporates a CNN for image data and a RN similar to the one used for run 2 for the metadata. The CNN is a ResNet-18 [7] pre-trained on ImageNet, but its last layer was replaced by a fully-connected layer of 512 units. The f network of the RN uses the same architecture from run 2, except its last layer is replaced by a fully-connected layer of 256 units. Then, the output of both the CNN and RN are concatenated into a single vector, followed by a fully-connected layer of 512 units and finally a single sigmoid unit for classification. The network is then trained as whole, with no specific tuning for handling the pre-trained weights.

Run 4: We proposed an alternative solution for run 1. We split the training images into 5 disjoint subsets, which are combined by taking 4 out of 5, covering all possible combinations of the 5 sets (similar to a 5-fold cross validation). This process results in 5 distinct and combined sets (each one composed of 4 subsets), which are used to fine-tune 5 independent GoogLeNets. In the prediction phase, we average the scores of being a flooding event given by each network and then rank the test set with them.

Run 5: We used a metadata-based approach based on an IR ranking solution, ranking test samples based on their estimation

MediaEval’17, 13-15 September 2017, Dublin, Ireland

of being flood. That estimation comes from a metric evaluation over a ranked list for each test sample, computed as follows. Let $D = \{d_1, d_2, \ldots\}$ be the dev set, and $T = \{t_1, t_2, \ldots\}$ be the test set. Also let $\langle M, F \rangle$ be a pair of a representation model M and a distance function F, in which M is applied over a sample s (the query from an IR perspective) and produces $M(s)$, and F is applied over a pair of samples previously modeled by M, so $F(M(s_1, s_2)) = f_{s_1,s_2}$ corresponds to the distance of s_1 to s_2 with respect to M and F. With $\langle M, F \rangle$ and a test sample t, we generate a ranked list $r(t)$ that contains up to $|D|$ pairs of $\langle d_i, f_{t,d_i} \rangle$, where $d_i \in D$, whose pairs are sorted by f_{t,d_i}.

We tested $\langle M, F \rangle$ pairs, then selected the ones who performed best on dev set. For the three best pairs, we produce three ranked lists for a sample. We use a graph-based rank-aggregation technique to produce an unified ranked list. By applying the same procedure to the dev samples as well, as if they were also queries, we generate graphs that combine their ranked lists, thus we end up with graphs for every sample. Given a test graph, we compare it to the dev graphs and produce a final ranked list. A graph-based dissimilarity function [12] is used to compare test and dev graphs. Given the set of ranked lists produced for each test sample, we estimate 'how much flood' a test sample is, using the NDCG@K measure [5]. The final submission file contains test samples decreasingly sorted by their estimation. We chose K by evaluating within the dev set, picking $K = 7$, which maximized the effectiveness.

The three best $\langle M, F \rangle$ pairs were $\langle$RF, WGU$\rangle$, $\langle$bigrams-TF, cosine$\rangle$, and $\langle$BoW-TF, cosine$\rangle$, where TF is a weighting functions, RF (relative frequency) is a graph-based text representation model [10], and WGU [12] is graph-based dissimilarity function.

3 FLOOD-DETECTION (FDSI)

For the **FDSI** task, we employed CNNs with dilated (or a-trous) convolution [4]. Unlike standard CNNs, networks composed of this type of convolution learn the given task by processing the input without downsampling it. This is only possible because dilated convolutions allow gaps (or "holes") inside their filters, which represent a great advantage in terms of computational processing, as well as in terms of learning, given that internal feature maps do not lose resolution (and information).

In this subtask, we proposed 4 CNNs. The most important one, which was exploited in all runs, is composed of 6 dilated convolution layers and the final fully-connected one, which is responsible for the classification. There are no poolings or normalizations inside the network. The first two convolutions have 5 × 5 filters with dilation rate 1. Convolutions 3 and 4 have 4 × 4 filters but larger rate 2. Finally, the last convolutions have smaller filters (3 × 3) but larger dilation rate 4. In a pre-processing stage, we normalized the images using the mean and standard deviation of each image band.

Run 1: We trained the aforementioned CNN by using overlapping patches of size 25 × 25 extracted from all training images. In the prediction phase, we also extracted overlapping patches with the same resolution from the testing images and averaged the probabilities outputted by the network.

Run 2: We processed the images exactly as in run 1 but using a larger patch, with 50 × 50 pixels, which tends to aggregate more context that could improve the learning process.

Run 3: We combined the features extracted from several distinct CNNs using a linear SVM. Specifically, the SVM receives as input features extracted from CNNs trained in run 1, 2, and 5, as well as: (i) a dilated CNN with pooling layers (but that do not reduce the resolution giving the padding), and (ii) two networks based SegNet [1] that uses deconvolution layers.

Run 4: We combined all networks presented in run 3 using a majority voting scheme.

Run 5: We trained a specific dilated CNN (using patches of 25 × 25) for each of the six locations, i.e., we had one network specialized for each location. The prediction is similar to run 1, except for the use of each CNN in its respective location. For the new location, we combined the features extracted from each CNN using a linear SVM, just like run 3.

4 RESULTS & DISCUSSION

Table 1 presents our results for the DIRSM subtask. The best results considering AP@480 was achieved by run 3 (95.84%), the neural network solution that combines textual and visual data. However, considering the MAP@[50,100,250,480] the visual only approach that combines results from 5 fine-tunned networks from GoogLeNet (91.59%) stood out (e.g., run 4). In both cases, the results of neural networks surpassed by far those yielded by the IR approach (run 5).

Table 1: Average Precision (%) at 480 and Mean Average Precision (MAP) (%) at cut-offs 50, 100, 250 and 480 (DIRSM).

	Run 1	Run 2	Run 3	Run 4	Run 5
AP@480	74.60	76.71	**95.84**	82.06	54.31
MAP@[50,100,250,480]	87.88	62.53	85.63	**91.59**	41.13

Our results for FDSI subtask (Table 2) indicated that the solution that extracts features using several CNNs and combines them with SVM produced the best results (run 3) for test items referring to either locations seen before in training set, or new locations.

Table 2: Mean Intersection Over Union (%) (FDSI).

	Run 1	Run 2	Run 3	Run 4	Run 5
Same locations	87.64	86.56	**88.23**	78.06	87.93
New locations	82.53	80.25	**84.10**	49.80	84.10

5 FINAL REMARKS & FUTURE WORK

As future work, for the DIRSM subtask, we intend to: (i) combine different networks (VGG, AlexNet, ResNet, DenseNet) and (ii) use RNs with open vocabulary models, as the current approach has its vocabulary limited to the training data. For the FDSI subtask, we intend to: (i) explore different learning algorithms as post-processing, and (ii) combine distinct networks.

ACKNOWLEDGMENTS

We thank FAPESP, FAPEMIG, CNPq, and CAPES.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. (2015). Preprint at http://arxiv.org/abs/1511.00561.

[2] Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python.* O'Reilly Media Inc.

[3] Benjamin Bischke, Patrick Helber, Christian Schulze, Srinivasan Venkat, Andreas Dengel, and Damian Borth. The Multimedia Satellite Task at MediaEval 2017: Emergence Response for Flooding Events. In *Proc. of the MediaEval 2017 Workshop* (Sept. 13-15, 2017). Dublin, Ireland.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. (2016). Preprint at http://arxiv.org/abs/1606.00915.

[5] M.-L. Fernández and G. Valiente. 2001. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters* 22, 6 (2001), 753–758.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT Press.

[7] K He, X Zhang, S Ren, and J Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 770–778. https://doi.org/10.1109/CVPR.2016.90

[8] Sandro Martinis, André Twele, and Stefan Voigt. 2009. Towards operational near real-time flood detection using a split-based automatic thresholding procedure on high resolution TerraSAR-X data. *Natural Hazards and Earth System Sciences* 9, 2 (2009), 303–314.

[9] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. (June 2017). Preprint at http://arxiv.org/abs/1706.01427.

[10] Adam Schenker, Horst Bunke, Mark Last, and Abraham Kandel. 2005. *Graph-Theoretic Techniques for Web Content Mining.* World Scientific Publishing Co., Inc., NJ, USA.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *Computer Vision and Pattern Recognition (CVPR).*

[12] W. D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray. 2001. Graph distances using graph union. *Pattern Recognition Letters* 22, 6 (2001), 701–704.