Assessing Flood Severity from Georeferenced Photos

Jorge Pereira
jorge.m.s.pereira@tecnico.ulisboa.pt
INESC-ID and Instituto Superior Técnico
Lisbon, Portugal

João Monteiro
joao.miguel.monteiro@tecnico.ulisboa.pt
INESC-ID and Instituto Superior Técnico
Lisbon, Portugal

Jacinto Estima
jacinto.estima@gmail.com
INESC-ID and Universidade Europeia
Lisbon, Portugal

Bruno Martins
bruno.g.martins@tecnico.ulisboa.pt
INESC-ID and Instituto Superior Técnico
Lisbon, Portugal

ABSTRACT

The use of georeferenced social media data in disaster and crisis management is increasing rapidly. Particularly in connection to flooding events, georeferenced images shared by citizens can provide situational awareness to emergency responders, as well as assistance to financial loss assessment, giving information that would otherwise be very hard to collect through conventional sensors or remote sensing products. Moreover, recent advances in computer vision and deep learning can perhaps support the automated analysis of these data. In this paper, focusing on ground-level images taken by humans during flooding events, we evaluate the use of deep convolutional neural networks for (i) discriminating images showing direct evidence of a flood, and (ii) estimating the severity of the flooding event. Considering distinct datasets (i.e., the European Flood 2013 dataset, and data from different editions of the MediaEval Multimedia Satellite Task), we specifically evaluated models based on the DenseNet and EfficientNet neural network architectures, concluding that these models for image classification can achieve a very high accuracy on both tasks.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision tasks**; **Neural networks**; *Visual content-based indexing and retrieval*; Supervised learning.

KEYWORDS

Flood Detection, Analysis of Georeferenced Images, Deep Learning for Computer Vision, Convolutional Neural Networks

ACM Reference Format:
Jorge Pereira, João Monteiro, Jacinto Estima, and Bruno Martins. 2019. Assessing Flood Severity from Georeferenced Photos. In *13th Workshop on Geographic Information Retrieval (GIR'19), November 28–29, 2019, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3371140.3371145

GIR'19, November 28–29, 2019, Lyon, France

ACM ISBN 978-1-4503-7260-2/19/11…$15.00
https://doi.org/10.1145/3371140.3371145

1 INTRODUCTION

The widespread use of mobile consumer electronics has made the act of taking and sharing photos online, for instance using smartphones or digital cameras coupled with GPS receivers, become commonplace. Information crowdsourcing through publicly shared photos has created a new opportunity to collect vast amounts of georeferenced image data, which can, for instance, be useful in the context of disaster and crisis management.

While crowdsourcing images posted on social media has been investigated in various contexts, leveraging such data in connection to flooding events (e.g., for detecting the extent of the flooding or the water level in inundated areas) remains relatively unexplored, apart from a few recent initiatives [4, 29]. This new data collection methodology can have the advantage of providing a local perspective on inundations, whereas previous studies mostly relied on remotely sensed data from an overhead perspective [2, 3, 14, 23, 25, 26, 29, 34]. The local detail of crowdsourced images can perhaps provide useful information for estimating the boundaries of flood-water, including partial blockage of roadways due to flooding [4, 5, 24, 33], which is important and otherwise very hard or impossible to achieve with conventional sensors. Crowdsourced georeferenced images can also be used to complement other information in flood monitoring systems (e.g., platforms such as the European Flood Awareness System[1]), through Geographic Information Retrieval (GIR) methods for selecting relevant and representative images in connection to particular flooding events [22].

This paper explores the use of deep learning approaches for image classification, in the context of flood detection. We relied on existing datasets of flood-related images collected from social media platforms (e.g., data from the MediaEval Multimedia Satellite Task [4, 5], and the photos made available in the context of the European Flood 2013 Dataset[2]), extending the ground-truth annotations associated to these images in order to discriminate between three distinct flood severity classes (i.e., not flooded, water level below 1 meter, and water level above 1 meter). Leveraging the aforementioned data (i.e., a total of 10,734 annotated images, which we also made available online[3]), we evaluated the application of recently proposed convolutional neural network architectures, specifically the DenseNet [13] and the EfficientNet [32] models, on different flood detection tasks, namely (i) discriminating photos showing

[1] http://www.efas.eu
[2] http://www.inf-cv.uni-jena.de/Research/Datasets/European+Flood+2013.html
[3] http://www.github.com/jorgemspereira/Classifying-Geo-Referenced-Photos

Table 1: Official results of the teams participating in the DIRSM sub-task of the MediaEval 2017 Multimedia Satellite Task.

	Visual		Metadata		Visual+Metadata	
Team	AP@480	AP@{50, 100, 250, 480}	AP@480	AP@{50, 100, 250, 480}	AP@480	AP@{50, 100, 250, 480}
MultiBrasil	74.60	87.88	**76.71**	62.53	**95.84**	85.63
WISC	50.95	62.75	66.78	74.37	72.26	80.87
CERTH-ITI	**87.82**	92.276	36.10	39.90	68.57	83.37
BMC	15.55	19.69	12.37	12.46	12.20	11.93
UTAOS	85.94	95.11	25.88	31.45	54.74	68.12
RU-DS	51.46	64.70	63.70	**75.74**	73.10	85.43
B-CVC	68.40	70.16	61.58	66.38	81.60	83.96
ELEDIA@UTB	77.62	87.87	57.07	57.12	85.41	90.39
MRLDCSE	86.81	**95.73**	22.83	18.23	85.73	92.55
FAST-NU-DS	64.88	80.98	65.00	71.79	64.58	80.84
DFKI	86.64	95.71	63.41	77.64	90.45	**97.40**

direct evidence of a flooding event, as proposed in the MediaEval 2017 Multimedia Satellite Task [4], and (ii) associating images to one of three flood severity classes. The obtained results suggest that deep learning methods for image classification can achieve a very high accuracy on both tasks, thus having a clear potential to complement other sources of georeferenced information (e.g., satellite imagery) related to flooding events.

The rest of this article is organized as follows: Section 2 presents previous research in the area. Section 3 describes the collections of photos used to support the evaluation experiments, detailing the process of extending the ground-truth annotations and presenting a statistical characterization of the resulting data. Section 4 presents the deep learning methods that were considered, specifically detailing the model adaptations and the considered training strategy. Section 5 presents the evaluation methodology and the obtained results. Finally, Section 6 summarizes our conclusions and discusses possible directions for future work, focusing on ongoing efforts that explore geo-spatial coordinates in connection to the images.

2 RELATED WORK

While extracting inundation levels from georeferenced crowdsourced images is a relatively new idea, some relevant studies were already presented and surveyed in the literature [29].

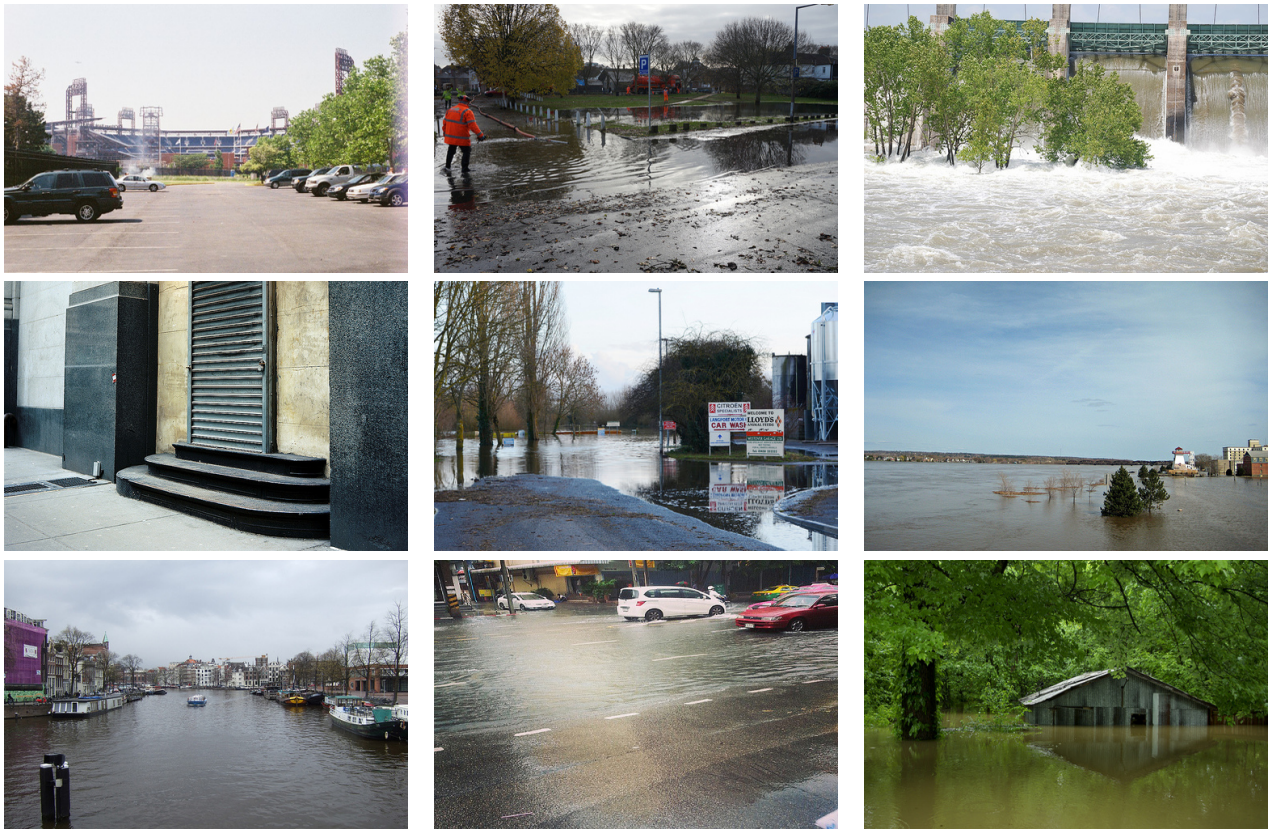
In the context of the Disaster Image Retrieval from Social Media (DIRSM) sub-task of the MediaEval 2017 Multimedia Satellite Task [4], a variety of different methods were proposed for detecting photos showing direct evidence of a flooding event. The MediaEval 2017 Multimedia Satellite Task also featured a sub-task concerned with segmenting flooded regions in satellite imagery and, in the 2018 edition [5], the goals were related with assessing road passability (in relation to the water level and the surrounding context) in both ground-level photos and satellite images. In the 2017 edition, the participants of the DIRSM task had access to a set of Flickr images with accompanying metadata (e.g., date, title, coordinates, a textual description, tags, etc.), and they were encouraged to explore the use of visual contents, metadata elements, or a combination of both. Systems were assessed on their capacity to retrieve flood-related images, and the results were measured in terms of the Average Precision (AP) at the cutoff threshold of 480 images, or in terms of the mean value for the AP at different cutoffs (i.e., 50, 100, 250 and 480 photos). Table 1 summarizes the results of the participating teams. When using visual contents alone, the best approaches (i.e., 87.82% in terms of AP@480 for team CERTH-ITI, and 95.73% in terms of AP@{50,100,250,480} for team MRLDCSE) corresponded to combinations of multiple features (i.e., MRLDCSE used an SVM classifier combining features extracted with AlexNet models pre-trained on the ImageNet and Places365 datasets, and CERTH-ITI used an SVM classifier combining pre-computed features provided by the task organizers, with features extracted with a GoogleNet model also pre-trained on the ImageNet dataset).

Geetha et al. proposed a method to measure water extents from images depicting individuals during flooding events [8]. The method uses colour-based thresholds to segment water in a given image, and it considers a water depth assessment method that first applies an accurate algorithm for face detection in the images, and then detects torso, waist, knee, and feet segments. The detection of body segments submerged in water can be used to assess the severity (i.e., the water depth) of the flood extent, although the authors only reported on initial evaluation results for the task of discriminating between flood and non-flood scenes.

Authors like Liu et al. [18] or Lo et al. [19], and ongoing activities such as the FLOODvision project[4], have discussed the use of Closed-Circuit Television (CCTV) signals to monitor water levels and associated spatio-temporal information, in the context of flood warning systems. The CCTV images can be processed to segment water bodies and, given that these images are collected from fixed positions, the water level can be detected by comparing virtual markers with carefully placed markers/rulers of known size.

Witherow et al. proposed an image processing pipeline for detecting the floodwater extent (i.e., for identifying image pixels corresponding to flooded areas) on georeferenced photos depicting inundated roadways, from image data captured by smartphones [33]. The proposed method is based on aligning and comparing pairs of images depicting dry versus flooded scenes, over the same location. First, the images go through a set of pre-processing operations consisting of (i) vehicle detection through a Region-Based Convolutional Neural Network (R-CNN) segmentation model [27], (ii) water edge detection through an approach based on the Hough transform, and (iii) image in-painting to remove the vehicles detected by the

[4] http://www.eawag.ch/en/department/sww/projects/floodvision/



(a) None (b) Below 1 Meter (c) Above 1 Meter

Figure 1: Samples from the final dataset, containing images labeled according to one of three flood severity classes.

R-CNN. Then, the images are converted to grayscale, and Scale-Invariant Feature Transform (SIFT) features are extracted. The SIFT features are used to match the image of the flooded scene against an image for the same region under a dry condition, and the dry condition image is warped in the registration process. After registration, pixel-wise subtraction yields the difference between the warped image and the flooded condition image. Next, Gaussian low-pass filtering and dilation operations are applied over the result of the subtraction, followed by a threshold-based segmentation operation that identifies the flooded and non-flooded areas.

Narayanan et al. presented another approach for urban flood monitoring using computer vision [24], in which crowdsourced georeferenced photos depicting partially submerged static structures (e.g., buildings, lampposts, etc.) are classified regarding the flood level. The geo-spatial coordinates associated to the images under analysis are used to search, on Google Street View, for images of the same places. SIFT features are used to compare the received photo depicting a flood against the retrieved reference images, and for selecting the reference image with an orientation that better matches the photo. Then, the Random Sample Consensus (RANSAC) algorithm is used to compute an homography matrix relating the obtained image with the test image. Manual inspection is used to define a line segmenting the submerged area in the photo and, finally, a homographic transformation of the flood line coordinates is used to project this information into the reference image, from which the flood height is derived (i.e., by analyzing the height of the part of a building that is submerged, and crossing this information with building height data from public records).

Giannakeris et al. described steps towards the development of a warning system, capable of detecting people and vehicles in danger over crowdsourced images [9]. In the proposed pipeline, a VGG-16 convolutional neural network, pre-trained on the Places365 dataset, is first used to classify images according to 3 classes of emergency events (i.e., fire, flood, or other). Next, a fully-convolutional network is used to segment the regions of the images where fire and flood pixels exist, for images belonging to the first two classes of emergencies. Separately, an R-CNN model is used to retrieve bounding boxes for persons and vehicles appearing in the images, and then these bounding boxes are classified as *safe targets*, *targets possibly in danger*, or *targets in danger*, according to the proximity towards the image segments corresponding to fire or flood pixels. The results are finally used to produce a qualitative risk assessment for the entire image, according to a three-level scale (i.e., low, medium, and high risk). Each component of the pipeline was evaluated separately and, in the case of the image classification component, the authors also report on the results (i.e., an accuracy of 97.50%) for 10-fold cross-validation with the training split of the dataset from the DIRSM sub-task of the MediaEval 2017 Satellite Task.

Deep learning-based flood detection approaches have also been studied by Lopez-Fuentes et al. [20, 21], which proposed to monitor

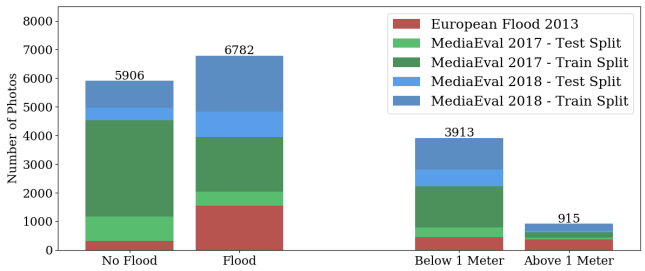


Figure 2: Statistical characterization for the final dataset.

river water levels by processing data collected from surveillance cameras installed near riverbeds. The authors compared three different deep neural network architectures in a segmentation task corresponding to the identification of pixels corresponding to water [20]. In the context of the DIRSM sub-task of the MediaEval 2017 Multimedia Satellite Task, the same authors proposed a multimodal system that consisted of a CNN to process visual data (i.e., an InceptionV3 model pre-trained on the ImageNet dataset) and a bi-directional Long Short-Term Memory (LSTM) network to extract semantic features from text metadata (i.e., titles, descriptions and tags) associated to the images [21]. The authors achieved interesting results on the MediaEval competition (i.e., the B-CVC team in Table 1), and combining both sources of information (i.e., text and visual contents) resulted on a value of 81.6% for the average precision at the cutoff value of 480 images, and on a value of 83.96% for the MAP at cutoffs 50, 100, 250 and 480.

3 AN IMAGE DATASET FOR EVALUATING FLOOD SEVERITY ESTIMATION

The photos that were used to support the experiments reported on this paper were originally made available as part of different datasets for evaluating computer vision and information retrieval experiments in tasks related to processing flood-related imagery.

The first of three different datasets that were considered in our work consists of 6,600 Flickr images extracted from the Yahoo Flickr Creative Commons 100 Million (YFCC100m) dataset, originally shared under Creative Commons licenses and made available in the context of the Disaster Image Retrieval from Social Media (DIRSM) sub-task of the MediaEval 2017 Multimedia Satellite Task [4]. Only one image per user was considered in the creation of this dataset, to avoid a bias towards content from the same locations and from the most active content-sharing users. Relevance scores were collected from two annotators and final ground truth labels, for whether the photos depict flood related information or not, were determined through the agreement of both annotators in classifying the images with a high confidence. The dataset was originally separated with a ratio of 80/20 into training and testing splits.

The second set of images came from the European Flood 2013 dataset, originally developed in the context of interactive content-based image retrieval experiments at the Computer Vision Group of the University of Jena. The majority of the images in this second dataset relate to the central European floods occurred in May and June 2013, and have been fetched in July 2017 from the Wikimedia Commons category named *Central Europe floods, May/June 2013*, or from its sub-categories. All images in the dataset were annotated by hydrologists regarding their relevance in terms of (i) depicting a flooding event, or (ii) containing visual cues (e.g., traffic signs) that can be used to derive an estimation of the inundation depth from the image. After manual inspection, we noticed that some images annotated with the indication of depicting a flooding event were too similar. To avoid any sort of bias, in our experiments, we decided to use only different images from the subset that contained visual clues to infer the water depth. From the complete set of 3,710 images, only 1,876 images were used in our experiments.

To further increase the set of images used in our tests, and given the similarity between the tasks of different editions of the MediaEval Satellite Task, we also used images from the Flood Classification for Social Multimedia sub-task of MediaEval 2018 [5]. The goal of the competition was to develop an algorithm that, given a set of images from social media, (i) retrieves all the images that provide evidence for road passability, and (ii) discriminates between images showing passable versus non passable roads. The original dataset for this task consisted of 11,070 identifiers that point to Twitter messages with images containing in the description the tags *flooding*, *flood* and/or *floods*. The majority of the images have been collected during three big hurricane events in 2017 and were annotated according to two dimensions: (i) one binary label for the evidence of road passability, and (ii) for those images that are labeled as showing evidence, a second binary label for the actual road passability classification. The images were labelled by human annotators through crowdsourcing. In order to use the images in our study, we manually re-labeled them regarding the containment of direct evidence of a flooding event. During this process, some near-duplicates or images with too poor resolution were discarded, resulting in a total of 4,212 images.

In the European Flood 2013 dataset, although the images were annotated according to containing elements from which an inundation depth could be derived, the actual depth was not considered as part of the expert annotations. To support the realization of experiments concerned with a more thin-grained characterization of the images, we decided to extend the ground-truth annotations associated to all the photos from the three different datasets, in order to discriminate between three distinct flood severity classes (i.e., not flooded, water level below 1 meter, and water level above 1 meter). A total of 1,954 images could not be annotated according to one of the aforementioned thin-grained classes due to (i) a lack of architectural features that could be used to estimate the water depth in an urban area, or (ii) some ambiguities in the height of the objects contained in the images (e.g., structures like bridges with pillars almost entirely submerged were present in the images, but without in-depth knowledge about the architectural infrastructures being depicted, it was impossible to estimate the water depth). In the case of multiple submerged objects that could lead us to assume that the flood had different depths at different parts of the terrain being depicted, the object closest to the point where the photograph was taken was used as the tiebreaker. In total, the re-annotation process resulted in a collection of 10,734 photos.

Figure 1 shows example images, together with the corresponding class assignments. In turn, Figure 2, presents a statistical characterization of the resulting dataset of 10,734 images, separately counting

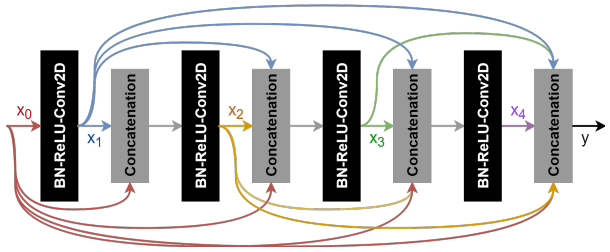


Figure 3: Graphical representation for a dense block.

the number of images in each class, and also counting images according to the different provenance sources (i.e., the training and testing splits from MediaEval 2017 and MediaEval 2018, and the European Flood 2013 dataset). It is interesting to notice that the per-class distribution of the number of images is somewhat skewed, with relatively few photos within the class for severe flooding events. The distribution of the instances per dataset is also unbalanced.

The experiments reported in Section 5 used different splits of the data, relying on the original MediaEval 2017, MediaEval 2018 and European Flood 2013 datasets for tests concerned with detecting flood-related images (e.g., using the original training/testing splits from MediaEval 2017, extending the set of training images that were originally available with images from the other datasets, or leveraging cross-validation on the complete set of images), and using the new complete (i.e., final) dataset, with thin-grained annotations, for the tests concerned with estimating flood severity.

4 THE IMAGE CLASSIFICATION METHODS

This section details the image classification methods used in our experiments. First, Section 4.1 presents the convolutional neural network architecture commonly referred to as DenseNet [13]. Then, Section 4.2 presents the more recent EfficentNet [32] architecture. Finally, Section 4.3 presents all the training and hyper-parameter tuning strategies that were considered in the experiments.

4.1 The DenseNet Neural Architecture

Convolutional Neural Networks (CNNs) have been extensively used in image classification tasks [35], processing the pixels (e.g., the RGB values) from input images through a composite of convolution and pooling operations, in order to obtain high-level features that inform final prediction layers – see Khan et al. for a survey on the topic [16], as well as an introduction to fundamental concepts related to the use of CNNs in image classification. Since very early, researchers observed that increasing the number of hidden layers in CNNs (i.e., the number of convolution and pooling operations) often leads to improvements. However, this increase can also raise several problems, including the vanishing of information between the distant layers during training. Recent work has shown that CNNs can be substantially deeper, more accurate, and efficient to train, if they contain shortcut connections between distant layers [36].

One of the earliest CNN architectures which used the idea of shortcut connections was ResNet [11]. In this neural model, information preservation occurs explicitly through additive identity transformations between subsequent layers (i.e., ResNet models use shortcut connections in which we sum the output feature maps of one layer with the corresponding incoming feature maps). However, some studies demonstrated that many ResNet layers contribute very little to the final output, and the number of parameters can be quite large. To address these issues, more recent work by Huang et al. proposed the idea of dense connectivity, presenting DenseNets [13].

The DenseNet architecture is built using multiple dense blocks, each as shown in Figure 3. Each of the dense blocks can be seen as a small CNN where each layer is connected to every other layer in a feed-forward fashion. The elementary operations underneath dense blocks, besides concatenations, correspond to pre-activation Batch Normalization (BN), followed by a ReLU activation [1] and then a 3×3 convolution operation. Whereas traditional CNNs with l layers have l levels of connections (i.e., between each layer and the subsequent layer), each dense block has $l \times (l+1)/2$ direct connections. This dense connectivity pattern presents several advantages, for instance addressing vanishing gradients, strengthening feature propagation, encouraging feature reuse, and even decreasing the number of parameters. The reduction on the number of parameters comes from the fact that there is no need to re-learn redundant feature maps, since the feature maps learned by any layer of a dense block can be accessed by all subsequent layers.

In a complete DenseNet, as shown in Figure 4, the input image is first processed through a 7×7 convolutional layer with a stride of 2, followed by a 3×3 maximum pooling operation, also with a stride of 2. The result is then passed to a sequence of 4 dense blocks interleaved with transition layers, and finally processed through a 7×7 global average pooling operation before the final output layer. The transition layers are responsible for improving the model compactness, first applying a batch normalization, and then a 1×1 convolution followed by an average pooling operation over all the feature maps produced by the dense block. Representing as m the number of feature maps of a certain dense block, and as θ the compression factor associated to the average pooling operation, a transition layer generates $\lfloor \theta \times m \rfloor$ feature maps as output.

Huang et al. compared the DenseNet and ResNet architectures on multiple datasets, including ImageNet [7]. DenseNets obtained results on par with ResNets, whilst requiring significantly fewer parameters and computation to achieve comparable performance.

4.2 The EfficientNet Neural Architecture

In recent work, Tan et al. [32] argued that CNN architectures should be scaled up in multiple dimensions for optimal performance, since scaling in only one direction (i.e., depth only) would result in rapidly deteriorating gains relative to the increases in terms of computation costs. Most of the commonly used CNN architectures are scaled up by adding more layers (e.g., DenseNet architectures can be scaled up to versions DenseNet-121, DenseNet-169, and DenseNet-201). Typically, the bigger the number (i.e., the number of blocks/layers on the network), the bigger the ability for the network to model a problem and achieve better results. However, as the authors demonstrated, simply going deeper rapidly saturates the gains (e.g., a DenseNet-1000 will not be much more accurate than a DenseNet-201). The other alternatives are to scale up the networks in width and resolution, but associated benefits quickly disappear as well. To address this problem, Tan et al. proposed a method to efficiently scale up CNNs. The method, referred to as compound scaling, uses a compound coefficient ϕ to uniformly scale the width, depth, and

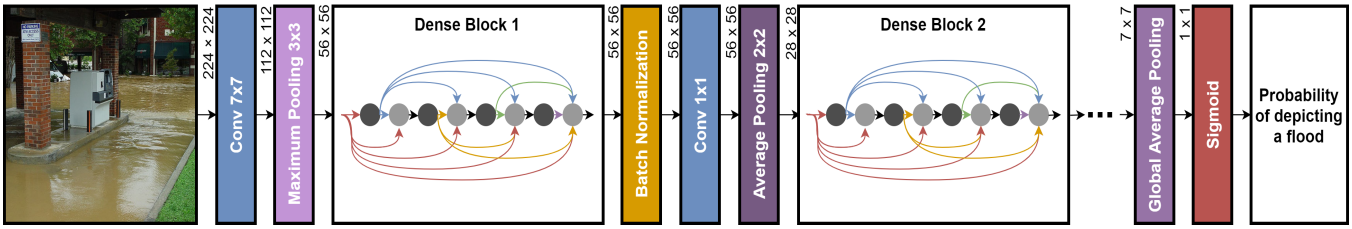


Figure 4: Graphical representation for the DenseNet architecture [13].

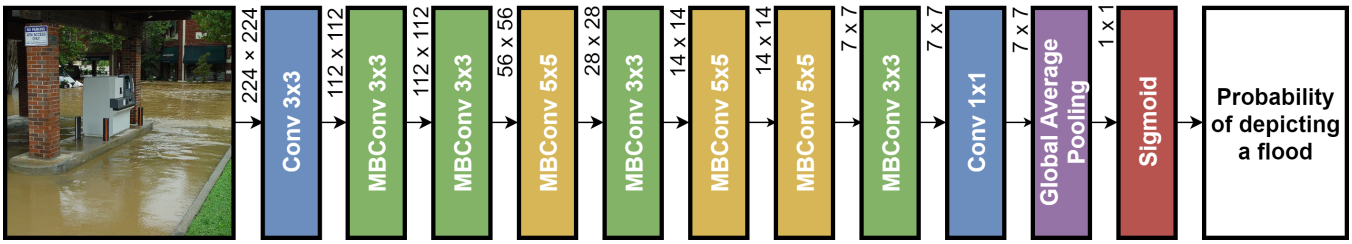


Figure 5: Graphical representation for the EfficientNet-B0 architecture [32].

resolution of a network, being formulated as follows:

$$\begin{aligned} \text{depth:} \quad & d = \alpha^{\phi} \\ \text{width:} \quad & w = \beta^{\phi} \\ \text{resolution:} \quad & r = \gamma^{\phi} \\ \text{s.t.} \quad & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ & \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{aligned} \tag{1}$$

In the previous expression, α, β, and γ are constants that can be determined by a grid search. Intuitively, ϕ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, and γ specify how to assign these extra resources to the network width, depth and resolution, respectively. This equation comes from the fact that a regular convolution operation is proportional to d, w^2 and r^2, meaning that doubling the network depth will double the computational cost, but doubling the width or resolution will increase this cost by four times. Since convolutional operations usually dominate the computational cost, scaling a CNN with Equation 1 will approximately increase the cost by $(\alpha \cdot \beta^2 \cdot \gamma^2)^{\phi}$. In the paper, the authors constrained this condition to be approximately equal to 2 such that for any new ϕ, the computational cost will approximately increase by 2^{ϕ}.

In order to test this compound scaling formula, the authors proposed a novel model referred to as EfficientNet. The main building blocks of the EfficientNet architecture are mobile inverted bottleneck convolution operations [28, 31], to which the authors also added squeeze-and-excitation optimization [12]. This elementary building block, named MBConv, essentially corresponds to a residual block that connects the beginning with the end through a skip connection. By adding these connections, the block gains the ability to assess earlier activations that were not modified by the convolutional operations. Each of these blocks starts with a 1×1 convolution in order to reduce the number of parameters, followed by a $n \times n$ depthwise convolution (i.e., a particular type of convolution operation where the kernel is divided into multiple kernels across the different channels in order to reduce the number of multiplications that are needed [6]), reducing even further the number of parameters when compared to typical convolution operations. Afterwards, a channel squeeze-and-excite operation is applied, which improves channel inter-dependencies at almost no computational cost [12]. The main idea is to add parameters in each channel of a convolutional block, so that the network can adaptively adjust the weighting of each feature map. A channel squeeze-and-excite operation starts by squeezing the global spatial information into a channel descriptor, followed by an excitation phase that aims to fully capture channel-wise dependencies. This operation is capable of learning a non-linear (and non-mutually-exclusive) relationship between the different channels. Finally, each MBConv ends with another 1×1 convolution to squeeze the feature map in order to match the initial number of channels.

The authors started by testing a model named EfficientNet-B0, for which we present a representation in Figure 5. Then, they scaled this model consecutively using the compound scaling method, in order to obtain a family of EfficientNets from B1 to B7. After applying the constrains specified in Equation 1 the authors discovered that the optimal balance consisted on 1.20 to the depth, 1.10 to the width and 1.15 to the resolution. This means that to scale up their B0 model, in order to keep it as efficient as possible while expanding the implementation and improving the accuracy, the depth of layers should increase by 20%, the width by 10% and the resolution by 15%. For the tests reported on this article, we decided to use the model corresponding to the EfficientNet-B3 architecture.

Tan et al. also compared EfficientNet models against other CNN architectures on the ImageNet dataset, with striking results when comparing the number of parameters/computations required versus almost every other CNN architecture (i.e., a 5 times reduction while keeping or beating most accuracy values). When comparing EfficientNet-B1 against DenseNet-264, the authors report a gain of 0.9% in top-1 accuracy over the ImageNet benchmark.

4.3 Training and Hyper-Parameter Tuning

Our experiments with DenseNet models used an implementation with 201-layers and $\theta = 0.5$, pre-trained on ImageNet and provided as part of the Keras[5] library. We also used a pre-existing Keras implementation for the EfficientNet model[6], together with pre-trained weights on ImageNet. Due to GPU memory constraints, and in order to obtain directly comparable results (i.e., without altering the batch size on our experiments), we used the EfficientNet-B3 since this was the biggest version that we could fit on our hardware. For fine-tuning the pre-trained models with our flood-related data, we considered a selection of hyper-parameters and model training strategies that relied on the guidelines from several previous publications, as for instance the guidelines from Xie et al. [35].

The last layer of the pre-trained DenseNet or EfficientNet models was replaced by a new fully-connected layer, with a number of nodes compatible with the classification task. In both scenarios that were considered, the entire set of network weights was afterwards fine-tuned with the flood-related images. When considering a binary classification (i.e., detecting images depicting a flooding event), the last layer consists of a single node with a sigmoid activation function, and training involves minimizing a binary cross-entropy loss. When estimating flood severity classes, the last layer consists of three output nodes, and training involves a softmax activation function together with the categorical cross-entropy loss function.

Training relied on the Adam [17] optimization algorithm, together with a cyclical learning rate [30]. In more detail, the learning rate varied between 10^{-5} and 10^{-4}, according to a triangular policy that decreases the cycle amplitude by half after each period (i.e., annealing the learning rate), while keeping the base learning rate constant. We used mini-batches of 16 images, created through a generator that considered simple real-time data augmentation procedures (i.e., randomly flipping the input images horizontally when providing them as input to the training algorithm, and/or randomly shifting the brightness by a factor between 0.8 and 1.2).

Training proceeded for up to a maximum of 50 epochs. However, a small validation set (i.e., 10% of each training split in cross-validation experiments, and 20% of all the available training data when using fixed splits) was used to define an early stopping criterion. Training stopped when the validation loss was not decreasing for 5 consecutive epochs. The final model weights were taken from the training epoch with the smallest value for the validation loss.

5 EXPERIMENTAL RESULTS

In a first set of experiments, we assessed the ability of the DenseNet and EfficientNet models to detect whether a given photo presents direct evidence of a flooding event, following the general task definition and evaluation methodology of the Disaster Image Retrieval from Social Media (DIRSM) sub-task of the MediaEval 2017 Multimedia Satellite Task. In the competition, the official metrics for evaluating participants, in terms of their ability to retrieve flood-related images, was the Average Precision at k (AP@k), considering a cutoff of $k = 480$, and also an average value across multiple cutoffs (i.e., 50, 100, 250, and 480). AP@k measures the number of relevant images among the top k retrieved results, thus taking the rank into consideration when sorting photos according to the confidence of the classifier in assigning the positive (i.e., flood-related) class. Besides evaluating results in terms of the AP@k metrics, we also measured the overall classification accuracy, as well as the standard precision, recall, and F1 metrics for the positive class. Table 2 presents the obtained results for the different models, trained through the complete procedure described in Section 4.3 (i.e., with hyper-parameters tuned to the best values).

The table features three main sets of rows, corresponding to (i) experiments in which results were evaluated on the official test split from the MediaEval 2017 competition, (ii) experiments based on a 10-fold cross-validation methodology, and (iii) the best results that were reported at MediaEval 2017 when using visual features alone. The first four rows, that are separated by horizontal lines, specifically correspond to using different sets of photos for model training, namely (i) all the photos in the official training split from MediaEval 2017, (ii) all the photos re-annotated from MediaEval 2018, (iii) all the photos in the European Floods 2013 dataset which contain objects that allow assessing the height of the water, and (iv) a combination of all the photos from the three previous items. The next four rows correspond to cross-validation experiments using (i) all photos from the training split of MediaEval 2017, (ii) all the photos re-annotated from MediaEval 2018, (iii) the same set of photos from European Floods 2013 dataset as in the previous set of rows, and (iv) a combination of the photos from MediaEval 2017 (i.e., train and test splits), MediaEval 2018, and from the European Floods 2013 dataset. Finally, the last row corresponds to the best results reported by the participants in MediaEval 2017, specifically in terms of the AP@480 metric, and the mean of the AP at the different cutoffs (i.e., 50, 100, 250, and 480).

The results confirm that all the considered models indeed achieve a very good performance in this particular task, significantly outperforming the official results from the competition. The best results over the test split from the dataset of the MediaEval 2017 competition, when using the official training split and when considering standard classification metrics, were obtained using the DenseNet model. However, when considering the ranked retrieval metrics used in the MediaEval competition, the best results were instead achieved by the EfficientNet model. In the cross-validation experiments, the EfficientNet model has achieved better results in all metrics, in the majority of the experiments. Nonetheless, the two models achieved very similar results, and future endeavors can perhaps consider experimenting with a larger EfficientNet model (e.g., the B4 variation), in an attempt to further improve the results.

In a second set of experiments, we assessed the ability of the same neural models to discriminate between the three different flood severity classes. These tests leveraged a 10-fold cross-validation methodology, using the entire set of photos that resulted from the annotation process described in Section 3. The quality of the results was measured in terms of macro-averaged values for the precision, recall, and F1 metrics. Besides these standard metrics for multi-class classification problems, we also measured the overall classification accuracy, and the Mean Absolute Error (MAE). In this last case, note that the different classes can be seen to correspond to ordinal values encoding the flood severity level (i.e., 0, 1 and 2), and thus we can measure the differences between the ground-truth and the estimated values through an error metric like the MAE.

[5] http://www.keras.io/applications/#densenet
[6] http://www.github.com/titu1994/keras-efficientnets

Table 2: Results on the task of discriminating photos showing direct evidence for a flooding event.

		Classification				Ranked Retrieval	
	Model	**Pre**	**Rec**	**F1**	**Acc**	**AP@{50,100,250,480}**	**AP@480**
MediaEval 2017 Train Split	DenseNet	**92.05**	91.66	**91.85**	**94.09**	99.49	98.26
	EfficientNet	88.74	**93.54**	91.08	93.33	**99.59**	**98.59**
MediaEval 2018 Re-annotated	DenseNet	67.52	**98.75**	80.20	82.27	**98.14**	**95.85**
	EfficientNet	**71.63**	97.29	**82.51**	**85.00**	96.65	91.73
European Floods 2013	DenseNet	**59.60**	93.13	**72.68**	**74.55**	**89.66**	**83.28**
	EfficientNet	56.19	**96.46**	71.01	71.36	85.60	79.76
All Photos	DenseNet	86.58	**95.42**	90.79	92.95	99.34	97.97
	EfficientNet	**90.02**	92.09	**91.04**	**93.41**	**99.52**	**98.30**
CV MediaEval 2017 Train Split	DenseNet	88.60	**89.26**	88.89	91.90	**99.78**	**99.32**
	EfficientNet	**90.56**	89.10	**89.80**	**92.64**	97.98	98.50
CV MediaEval 2018 Re-annotated	DenseNet	96.40	97.10	96.75	95.61	**99.98**	**99.90**
	EfficientNet	**98.06**	**97.56**	**97.81**	**97.06**	**99.98**	99.88
CV European Floods 2013	DenseNet	96.73	96.58	96.63	94.46	**100.0**	**99.99**
	EfficientNet	**98.26**	**97.22**	**97.73**	**96.27**	**100.0**	**99.99**
CV All Photos	DenseNet	94.69	**94.41**	94.54	93.96	99.98	99.99
	EfficientNet	**95.87**	94.25	**95.05**	**94.56**	**100.0**	**100.0**
Best MediaEval 2017	—	—	—	—	—	95.73	87.82

Table 3: Results for cross-validation experiments on the task of classifying photographs according to the three different flood severity classes, using different neural models and training strategies.

				Macro-Averaged		
	Model	**Acc**	**MAE**	**Pre**	**Rec**	**F1**
Complete approach	DenseNet	**90.46**	**0.103**	**85.16**	83.96	**84.87**
	EfficientNet	90.24	0.107	84.44	**84.10**	84.19
- data augmentation	DenseNet	88.97	0.120	**83.88**	81.75	82.53
	EfficientNet	**90.03**	**0.111**	**83.88**	**83.88**	**83.80**
- cyclical learning rate	DenseNet	88.93	0.122	**83.29**	80.95	82.21
	EfficientNet	**89.39**	**0.118**	83.08	**82.80**	**82.86**
- model pre-training	DenseNet	**90.18**	**0.108**	**85.07**	**83.87**	**84.28**
	EfficientNet	70.83	0.312	61.98	55.42	55.81
Only EU-Floods'13 data	DenseNet	84.17	0.169	84.87	85.65	84.50
	EfficientNet	**84.94**	**0.163**	**85.24**	**85.84**	**85.28**

Table 3 presents the obtained results, and Figure 7 depicts example images for each cell in a classification confusion matrix (obtained using the best model), again attesting to the high quality of the predictions returned by the different models. The 2nd, 3rd and 4th sets of rows, shown on Table 3, correspond to ablation tests in which we removed some of the strategies listed in Section 4.3 (i.e., not considering data augmentation procedures, random initialization instead of model pre-training with ImageNet, and a fixed learning rate of 10^{-5} instead of the cyclical scheme). The last row corresponds to a cross-validation test in which we only used photos from the European Flood 2013 dataset, and in which the flood-related images are known to contain objects from which the water depth can, perhaps, be derived.

The DenseNet model achieved the best results in this fine-grained classification task. It is interesting to notice that the experiments with the smaller European Floods 2013 dataset produced slightly inferior results, despite the fact that these images should, in principle, be more informative. This result, and also the fact that worse results are available for the class related to floods with water above 1 meter, suggests that the number of images available for model training can indeed impact the result quality.

6 CONCLUSIONS AND FUTURE WORK

This paper addressed the use of convolutional neural networks for analyzing ground-level images taken during flooding events, specifically in the tasks of (i) discriminating images showing direct evidence of floods, or (ii) estimating the severity of the flooding event in terms of three distinct classes. Considering distinct datasets (i.e., the European Flood 2013 dataset, and data from the 2017 and 2018 editions of MediaEval competitions focusing on flooding events),

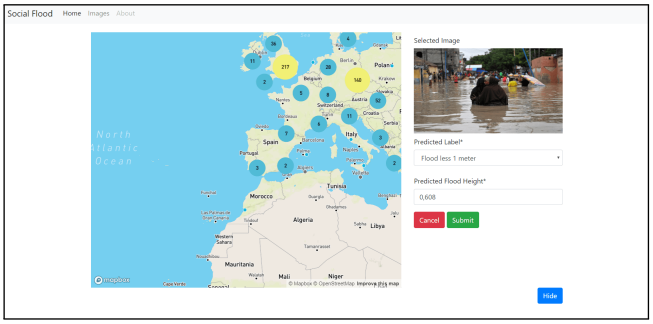


Figure 6: Screenshot of a web application for managing collections of georeferenced photos depicting flooding events.

we specifically evaluated models based on the DenseNet and EfficientNet neural architectures, concluding that these approaches can indeed produce high-quality classification results.

The classification models described in this paper have also been integrated in a prototype GIR application for managing collections of georeferenced photos depicting floods. Figure 6 provides a screenshot for this application, and the source code is available from a Github repository[7]. Users can upload photos, and search for photos in the indexed collection according to geo-spatial coordinates or other metadata elements. When uploading photos, our classification model estimates the water depth, and an heuristic procedure that leverages the geo-spatial location of the photo, together with the classification and a global high-resolution Digital Elevation Model (DEM), produces a more detailed estimate for the water level (i.e., we output the difference between the height, in meters, for the location of the photo, and the median height within a surrounding region that depends on the 3-class estimate for the water depth). Users are then encouraged to refine the automatically produced estimates, by comparing objects in the photo against references provided through the application (i.e., example images showing the typical heights associated to partially submerged persons, vehicles, or other types of common features with a known height).

Despite interesting experimental results, there are also many ideas for future work. As noted in connection to the prototype application, photos taken during floods often contain objects whose approximate dimensions are known (e.g. cars, bikes, traffic signs, architectural features of buildings, etc.). These objects, partially immersed, can serve as references for estimating the water level. For future work, in complement to end-to-end classifiers based on CNNs such as the ones reported on this paper, we would like to explore semantic segmentation models to infer the position and relative level of occlusion of particular types of objects (i.e., objects from which measurements can be derived) within images, in order to automatically produce a thin-grained estimate for the water level.

Different neural architectures can also be tested in the future, in an attempt to further improve the results. For instance, both Katharopoulos and Fleuret [15] and Guan et al. [10] proposed neural attention methods to guide the estimates and emphasize the parts of input images that are, perhaps, more useful for a certain classification. For future work, it would be interesting to also experiment with a similar method in our particular image classification problem, in an attempt to emphasize objects from which more precise estimates for the water level could be derived.

We are particularly interested in the development of approaches that can combine remote sensing and ground-level georeferenced imagery to support the assessment of flood severity, and there are many interesting ideas that can be explored in this direction (e.g., combine interpolated estimates produced with basis on the coordinates for the georeferenced photos classified as depicting floods, with estimates inferred from satellite imagery, in order to map flooded regions with a higher accuracy).

[7] http://www.github.com/jorgemspereira/Social-Flood

ACKNOWLEDGMENTS

This research was supported through Fundação para a Ciência e Tecnologia (FCT), specifically through the project grants PTDC/EEI-SCR/1743/2014 (Saturn), PTDC/CTA-OHR/29360/2017 (RiverCure), and PTDC/CCI-CIF/32607/2017 (MIMU), as well as through the INESC-ID multi-annual funding from the PIDDAC programme with reference UID/CEC/50021/2019. We also gratefully acknowledge the support of NVIDIA Corporation, with the donation of the two Titan Xp GPUs used in our experiments.

REFERENCES

[1] Abien Fred Agarap. 2018. Deep Learning using Rectified Linear Units (ReLU). *arXiv preprint 1803.08375* (2018).
[2] Kashif Ahmad, Konstantin Pogorelov, Michael Riegler, Nicola Conci, and Pal Halvorsen. 2017. CNN and GAN Based Satellite and Social Media Data Fusion for Disaster Detection. In *Proceedings of the MediaEval Workshop.*
[3] Benjamin Bischke, Prakriti Bhardwaj, Aman Gautam, Patrick Helber, Damian Borth, and Andreas Dengel. 2017. Detection of Flooding Events in Social Multimedia and Satellite Imagery using Deep Neural Networks. In *Proceedings of the MediaEval Workshop.* MediaEval.
[4] B. Bischke, P. Helber, C. Schulze, V. Srinivasan, A. Dengel, and D. Borth. 2017. The Multimedia Satellite Task at MediaEval 2017. In *Proceedings of the MediaEval Workshop.*
[5] B. Bischke, P. Helber, Z. Zhengyu, J. de Bruijn, and D. Borth. 2018. The Multimedia Satellite Task at MediaEval 2018. In *Proceedings of the MediaEval Workshop.*
[6] Francois Chollet. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*
[7] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*
[8] M. Geetha, M. Manoj, A .S. Sarika, M. Mohan, and S. N. Rao. 2017. Detection and estimation of the extent of flood from crowd sourced images. In *Proceedings of the International Conference on Communication and Signal Processing.*
[9] Panagiotis Giannakeris, Konstantinos Avgerinakis, Anastasios Karakostas, Stefanos Vrochidis, and Ioannis Kompatsiaris. 2018. People and vehicles in danger-A fire and flood detection system in social media. In *Proceedings of the IEEE Image, Video, and Multidimensional Signal Processing Workshop.*
[10] Q. Guan, Y. Huang, Z. Zhong, Z. Zheng, L. Zheng, and Y. Yang. 2018. Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification. *arXiv preprint 1801.09927* (2018).
[11] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*
[12] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*
[13] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*
[14] S. Andreadis E. Michail I. Gialampoukidis S. Vrochidis K. Avgerinakis, A. Moumtzidou and I. Kompatsiaris. 2017. Visual and textual analysis of social media and satellite images for flood detection. In *Proceedings of the MediaEval Workshop.*
[15] Angelos Katharopoulos and François Fleuret. 2019. Processing Megapixel Images with Deep Attention-Sampling Models. *arXiv preprint 1905.03711* (2019).

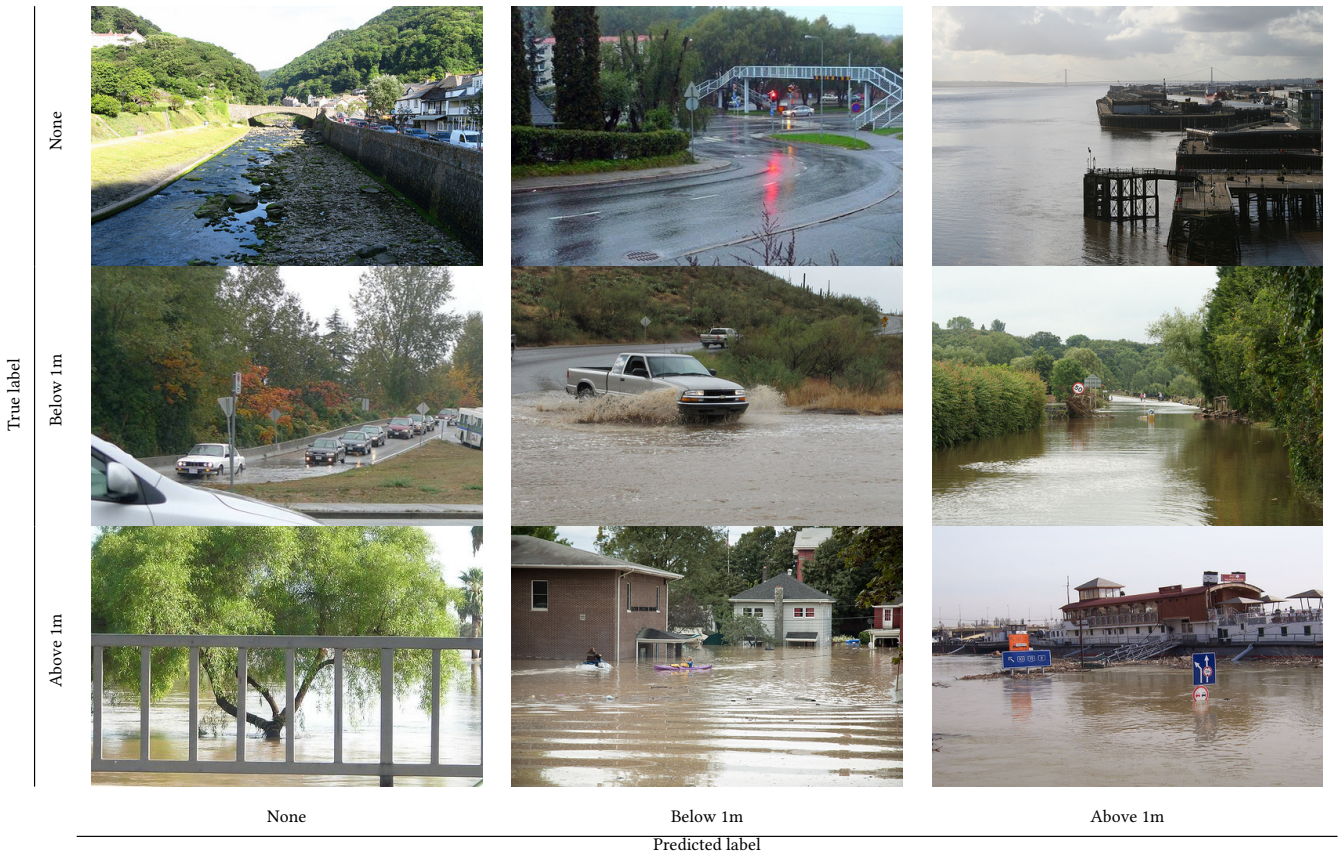


Figure 7: Example images depicting different flood severity classes, for each cell in a classification confusion matrix.

[16] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, and Mohammed Bennamoun. 2018. A guide to convolutional neural networks for computer vision. *Morgan & Claypool Synthesis Lectures on Computer Vision* 8, 1 (2018).

[17] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*.

[18] L Liu, Y Liu, X Wang, D Yu, K Liu, H Huang, and G Hu. 2015. Developing an effective 2-D urban flood inundation model for city emergency management based on cellular automata. *Natural Hazards and Earth System Sciences* 15 (2015).

[19] Shi-Wei Lo, Jyh-Horng Wu, Fang-Pang Lin, and Ching-Han Hsu. 2015. Visual sensing for urban flood monitoring. *Sensors* 15, 8 (2015).

[20] L. Lopez-Fuentes, C. Rossi, and H. Skinnemoen. 2017. River segmentation for flood monitoring. In *Proceedings of the IEEE International Conference on Big Data*.

[21] L. Lopez-Fuentes, J. van de Weijer, M. Bolanos, and H. Skinnemoen. 2017. Multimodal Deep Learning Approach for Flood Detection.. In *Proceedings of the MediaEval Workshop*.

[22] V. Lorini, C. Castillo, F. Dottori, M. Kalas, D. Nappo, and P. Salamon. 2019. Integrating Social Media into a Pan-European Flood Awareness System: A Multilingual Approach. In *Proceedings of the International Conference on Information Systems for Crisis Response and Management*.

[23] A. Zubiaga N. Tkachenko and R. Procter. 2017. WISC at Mediaeval 2017: Multimedia satellite task. In *Proceedings of the MediaEval Workshop*.

[24] R. Narayanan, L. Vm, S. Rao, and K. Sasidhar. 2014. A novel approach to urban flood monitoring using computer vision. In *Proceedings of the International Conference on Computing, Communication and Networking Technologies*.

[25] Keiller Nogueira, Samuel Fadel, Icaro Dourado, Rafael Werneck, Javier A. V. Muñoz, Otávio A. B. Penatti, Rodrigo Calumby, Lin Li, Jefersson A. Dos Santos, and Ricardo Torres. 2017. Data-Driven Flood Detection using Neural Networks. In *Proceedings of the MediaEval Workshop*.

[26] Jorge Pereira, Maria Dias, João Monteiro, Jacinto Estima, Joel Silva, João Moura Pires, and Bruno Martins. 2019. A Dense U-Net Model Leveraging Multiple Remote Sensing Data Sources for Flood Extent Mapping. Unpublished Technical Report.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the Conference on Neural Information Processing Systems*.

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[29] Linda M See. 2019. A Review of Citizen Science and Crowdsourcing in Applications of Pluvial Flooding. *Frontiers in Earth Science* 7 (2019), 44.

[30] L. N. Smith. 2017. Cyclical Learning Rates for Training Neural Networks. In *Proceeedings of the IEEE Winter Conference on Applications of Computer Vision*.

[31] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. 2019. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[32] Mingxing Tan and Quoc V. Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv preprint 1905.11946* (2019).

[33] Megan A. Witherow, Cem Sazara, Irina M. Winter-Arboleda, Mohamed I. Elbakary, Mecit Cetin, and Khan M. Iftekharuddin. 2018. Floodwater detection on roadways from crowdsourced images. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* (2018).

[34] L. Peng J. Z. Y. Yang X. Fu, Y. Bin and H. T. Shen. 2017. BMC at Mediaeval 2017 multimedia satellite task via regression random forest. In *Proceedings of the MediaEval Workshop*.

[35] J. Xie, T. He, Z. Zhang, H. Zhang, Z. Zhang, and M. Li. 2018. Bag of tricks for image classification with convolutional neural networks. *arXiv preprint 1812.01187* (2018).

[36] Linan Zhang and Hayden Schaeffer. 2018. Forward Stability of ResNet and Its Variants. *arXiv preprint 1811.09885* (2018).