International Journal of Geo-Information

MDPI

Article

Extraction of Pluvial Flood Relevant Volunteered Geographic Information (VGI) by Deep Learning from User Generated Texts and Photos

Yu Feng * and Monika Sester

Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Appelstraße 9a, 30167 Hannover, Germany; monika.sester@ikg.uni-hannover.de

* Correspondence: yu.feng@ikg.uni-hannover.de; Tel.: +49-511-762-19437

Received: 7 November 2017; Accepted: 21 January 2018; Published: 25 January 2018

Abstract: In recent years, pluvial floods caused by extreme rainfall events have occurred frequently. Especially in urban areas, they lead to serious damages and endanger the citizens' safety. Therefore, real-time information about such events is desirable. With the increasing popularity of social media platforms, such as Twitter or Instagram, information provided by voluntary users becomes a valuable source for emergency response. Many applications have been built for disaster detection and flood mapping using crowdsourcing. Most of the applications so far have merely used keyword filtering or classical language processing methods to identify disaster relevant documents based on user generated texts. As the reliability of social media information is often under criticism, the precision of information retrieval plays a significant role for further analyses. Thus, in this paper, high quality eyewitnesses of rainfall and flooding events are retrieved from social media by applying deep learning approaches on user generated texts and photos. Subsequently, events are detected through spatiotemporal clustering and visualized together with these high quality eyewitnesses in a web map application. Analyses and case studies are conducted during flooding events in Paris, London and Berlin.

Keywords: social media; crowdsourcing; volunteered geographic information; multimedia information retrieval; convolutional neural network; transfer learning; word embedding; flood mapping

1. Introduction

Flood, as one of the great disasters, endangers people's safety and their property. Generally, it can be categorized into three types, namely coastal flood, fluvial (river) flood and pluvial (rainfall) flood [1]. For the coastal and fluvial floods, tide and river gauges were built for monitoring water levels in real time, e.g., online river gauge maps for the UK and Ireland [2], tides and currents maps for the United States [3]. However, pluvial floods, which are normally caused by local, fast storm events with very high rainfall rates, are hard to be monitored and observed. Pluvial floods in urban areas are a great challenge for many cities. They are also expected to happen more frequently in the future [4]. These events may lead to failure of the drainage system of a city and have a high potential for damage, as evidenced by the flooding events in Beijing in June, 2012 [5] and in Berlin on 29 June, 2017 [6]. Even though the weather forecast may predict the amount of rainfall and weather sensor networks [7] may provide precipitation measurements in real time, the observation of the inundations or inlets overflow on the ground in real time is still hard to achieve. It is even harder for regions that lack meteorological infrastructure.

Crowdsourcing is a rapidly developing technique for event detection and is frequently used for domains such as public health and emergency response. Google Flu Trends [8] was a web service that tried to detect flu outbreaks based on users' search queries about flu symptoms. It was said to

provide predictions highly correlated with the actual flu outbreaks recorded by the US Centers for Disease Control and Prevention (CDC) [9]. However, overestimation of such events may sometimes happen [10]. Thus, accuracy and reliability are general issues for applications using crowdsourcing. For natural disaster events, first hand information from the people in the affected area is desirable for emergency management. Crowdsourcing is often used for collecting disaster relevant reports from voluntary users in real time. Some of these reports may contain user provided geographic information, thus they are regarded as Volunteered Geographic Information (VGI). Even though many applications nowadays for VGI are focusing on mapping (e.g., OpenStreetMap), potential of VGI used for early warning and emergency management has also been addressed by many researchers [11,12]. Lots of applications were already built to detect or analyze various disaster events based on VGI, such as earthquakes [13–15], floods [5,16,17], storms [18] and fires [19,20]. Since pluvial flood is one of the disasters that severely affects people and is normally directly caused by heavy rainfall events, a system is desirable to efficiently extract the voluntarily posted tweets relevant to rainfall and flooding and detect such events in real time.

The quality of the information retrieval plays a key role for event detection or further spatiotemporal analyses [21]. High quality topic relevant texts and photos can improve the situation awareness during the event and provide the decision makers with more detailed information in real time. The applications mentioned above have applied different kinds of approaches to retrieve disaster relevant tweets, mostly based only on the user generated texts. Methods such as keyword filtering [22,23] or classical Natural Language Processing (NLP) [24,25] were used to retrieve disaster related information. However, using only the text information leads to the retrieval of a large amount of false positive documents, which is due to the inherent ambiguities. Therefore, a retrieval based only on textual information is unlikely to be accurate.

Nowadays, most of the social media platforms allow users to share their photos (e.g., flickr, Instagram, Twitter). Photos can improve the situation awareness during a disaster event significantly. Therefore, photos were also used for event detection, but they are normally not automatically interpreted. For instance, the framework OEDIM [26] used keyword filtering to identify flood relevant social media messages. Additionally, it offers analysts the possibility to manually assess whether the retrieved photos are relevant to flood events or not, but has no automatic information extraction from the photos. Only recently, some researchers have applied deep learning techniques on 6600 Flickr images with metadata for multimedia flood relevant information retrieval [27,28], which have shown great potential for detecting flooding events based on real-time social media data. However, applications which include such techniques are not yet available as a service for cities.

In a similar spirit, the approach in this paper uses text processing techniques with deep learning. In addition to the texts, rainfall and flood scenarios are also identified by image recognition using deep learning as a separate independent source. Only with the confirmation from both texts and photos, VGI is considered as high quality eyewitnesses for rainfall and flood events. They are subsequently used as input for spatiotemporal clustering and hot spot detection to detect such events.

The information retrieved by our approach cannot only be used for detecting rainfall and flood events to improve the situation awareness, but can also be used for documenting such events. On the one hand it can be used to verify hydrological modelling results [29], on the other hand it can support the risk and loss analyses after the disaster. Currently, the risk and loss analyses depends greatly on telephone interviews. Tweets extracted by this approach could serve as a guidance for where the affected people are. Large-scale remote sensing flood mapping could also be significantly improved by only a small amount of VGI data [17]. In the near future, the functionalities offered in this paper will be embedded in an early warning system together with the predictions from meteorological and hydrological models. Furthermore, a similar framework can be established for other disaster events, which have significant features in both text and image, such as fire or lightning strokes.

The framework of our approach is shown in Figure 1. In our case, only the social media posts including both texts and photos are analyzed. Classifiers for both texts and images are trained and

applied separately and the individual evidences are combined. In the end, events are detected by spatiotemporal analysis. This paper is organized as follows. The next section is an introduction to social media data acquisition and storage. In Section 3, we introduce the related methods for interpreting flood relevant social media information. In Section 4, we describe the training of text classifiers and their comparisons based on an automatically annotated social media text dataset. In Section 5, we discuss the training of image classifiers and their comparison based on a manually annotated social media image dataset using a transfer learning approach. With the confirmation from both the text and image classifiers, high quality eyewitnesses of the rainfall and flood events are extracted, which used as individual hints for a possible event. In Section 6, spatiotemporal clustering is used to detect events and the daily hot spots are detected. In Section 7, the results of our application are described and visualized using a web application. In Section 8, we compare the results with an independent data source, namely precipitation and analyze the correlation between daily precipitation records and the daily amount of high quality eyewitnesses within 45 days in Paris and 14 days in London. In the last section, we conclude and give an outlook on future work.

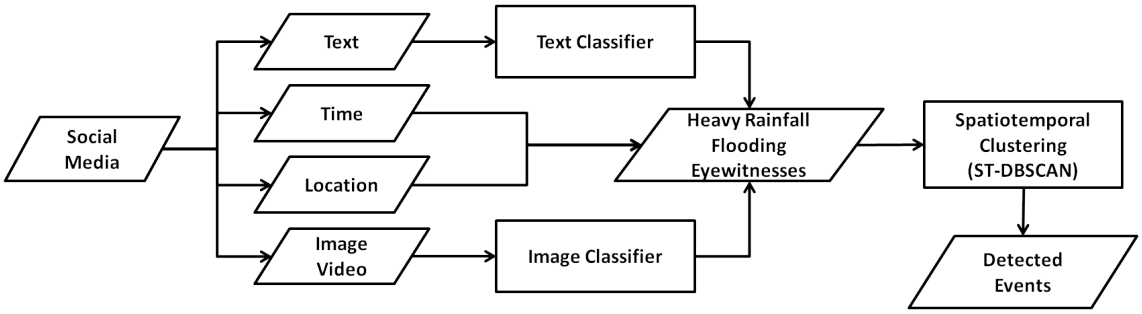

Figure 1. Work flow for pluvial flood relevant VGI extraction.

2. Social Media Data Acquisition

Since the aim is to detect rainfall and flooding events in real time, after comparing multiple social media platforms, Twitter was chosen for collecting pluvial flood relevant VGI. Currently, Twitter has 328 million active users worldwide [30], which leads to a large amount of user generated data. Because of its public Streaming API [31], we can access the real-time data streams of Twitter users. However, the number of tweets that can be crawled is restricted by the request limit of the Streaming API [32]. The API permits a pre-filtering according to geographical bounding box, keywords or languages. Therefore, instead of collecting tweets globally, a study area was defined from 24°32′47.4″ W to 18°30′ E in longitude and from 27°38′10.68″ N to 71°11′7.8″ N in latitude to collect geotagged Twitter data (as shown in Figure 2). Our study area covers most of the big cities in Western Europe. Additionally, we also filtered the data stream according to language and preserved only the tweets in seven frequently used languages within the study area, namely English, French, German, Italian, Spanish, Portuguese and Dutch. These are also the languages currently supported by the NLP tools used in our framework. At this step, no keyword filtering was applied to the Streaming API.

By restricting the area and filtering with respect to languages, the limitation by the Streaming API is greatly overcome, so that we achieve high completeness of the crowdsourcing data. Each tweet is downloaded as a json file and subsequently stored in a MongoDB database [33] since 15 May 2016. Many tweets may differ from each other with a different number of fields, therefore, the MongoDB database, as a NoSQL database [34] is ideal for this kind of data, as it does not require all the documents to have exactly the same fields.

Figure 2. Study area for collecting Twitter data (Basemap: Google Maps).

In general, only about 2% of the tweets are geotagged [35]. The proportion of the geotagged tweets with photos is shown in Figure 3. In this example, from 1 June 2016 to 30 June 2016, a total of 3.6 millions geotagged tweets were collected from 473,004 users. 59.1% of the geotagged tweets contain photos or references to photos on Instagram. The majority of these Tweet are shared Instagram posts with shortened text and URL link. Therefore, an extension to download these Instagram posts with full texts and images was also developed to improve the completeness of our collected VGI data.

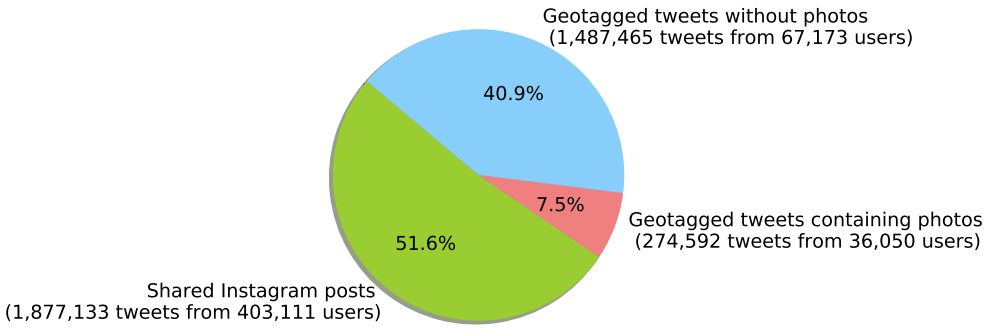

Figure 3. Proportion of geotagged tweets containing photos.

3. Related Methods for Interpreting Flood Relevant Social Media Information

With the rapid development of computer vision and NLP techniques in recent years, many studies nowadays are focusing on interpreting disaster relevant information from social media messages, especially using deep learning. Great achievements has been made for visual recognition ever since the proposal of Convolutional Neural Networks (ConvNets) [36]. The success of AlexNet [37] during the ImageNet challenge [38] in 2012 has also proved this method to be an efficient way to classify images. Training typically involves large training data sets, consisting of labelled example data. Since the features learned by a pre-trained model are transferable to other image recognition domains [39,40], transfer learning [41] is a popular approach for image classification with less training examples. As described in DECAF [39], it is possible to use a pre-trained ConvNet as feature generator and apply classical machine learning such as Support Vector Machine (SVM) or logistic regression to train a model with good performance. Transfer learning is utilized, such as classification of satellite images [42], vehicles detection based on RGB images or LiDAR data [43,44], visual floor count determination [45] or visual localization [46]. Only recently, this approach was used for retrieving flooding relevant social

media photos [27,28]. Most of the researches has been tested with only single methods such as SVM for transfer learning and no comparison with other alternatives methods are given.

Retrieval of flood relevant texts from social media has a much longer history than the retrieval of flood relevant photos. In many studies, in which social media is used as an information source for flood mapping or flood event detection, keyword filtering is frequently applied to extract the relevant information. Flood relevant keywords are filtered based on the social media texts. For instance, "hoch" and "wasser" were filtered for flood mapping in Germany, 2013 [22]. "Flood" and "Joaquin" were filtered for the South Carolina floods in 2015 [23]. For building the systems to detect various types of disasters, keyword lists were carefully collected for different disasters, different stages of disasters [47], and also for different languages [35]. The terms which cover the type of disaster (e.g., "flood"), its impact (e.g., "damage"), the perceptible triggers (e.g., heavy rain) were searched in [26]. Keyword lists are typically collected based on personal knowledge about the events. Simple keyword filtering to detect flood relevant information leads to the retrieval of a large amount of false positive examples.

Machine learning approaches have been applied to extract disaster related documents. An SVM trained on linguistic and statistical features was applied to detect earthquake relevant messages [13]. Latent Dirchlet allocation (LDA) was used to identify messages belonging to the topics relevant to flooding. With these messages, an SVM classifier was trained to make predictions about new messages [5]. With the idea similar to sentiment analysis, classical NLP methods using tf-idf (term frequency-inverse document frequency) [48] features were also utilized for extracting topic relevant documents. Text classification was applied for the analyses of disaster related tweets [24]. In our previous work, these methods were also used for retrieving rainfall and flooding relevant tweets [25].

With the development of deep learning methods for NLP in recent years, these methods are also being used for extracting disaster relevant messages. For tasks involving sentiment analysis over textual contents, ConvNets were utilized on the sentences [49], which were represented by word vectors generated by Word2Vec [50]. A similar approach was also applied on manually annotated social media texts in Chinese, which were collected from the Twitter-like social media platform Sina Weibo to extract earthquake relevant messages [51]. Their model was trained based on a balanced dataset with 2847 sentences containing the keyword "earthquake", where half of them were relevant to real earthquake events and the others not. For training this dataset, compared with SVM (85.4% in accuracy), ConvNets could achieve an accuracy of 91.6%, which indicates a significant improvement on classification accuracy.

Therefore, in this paper, we systematically tested both text and image classification methods from the literature, we used novel methods for automatic labelling and found optimal models which were embedded in our in our application for pluvial flood detection.

4. Interpretation of Social Media Texts

For the interpretation of social media texts, we first pre-processed raw texts collected by crowdsourcing and then labeled the texts automatically using historical rainfall records. Five classical NLP methods and one deep learning NLP method using word embedding are applied to train the text classifiers. After a systematic comparison, the model, which performs the best, is selected and embedded into our framework for further analyses.

4.1. Pre-Processing and Training Preparation

Since social media posts contain lots of noise, therefore, a pre-processing step is needed. Besides the raw text as the most relevant information, also the fields creation time, coordinates, source, media, user's screen name, language and text were used for the analyses. During the text pre-processing, the punctuation marks, numbers and URL were removed from the raw text. Since some of the emoticons are also related to our topic, the emoticons were not removed.

In NLP, reducing stop-words and stemming are standard techniques as pre-processing steps. Stopwords are the most common words in a language, such as article, pronouns or prepositions. Stemming is the process, which reduces each word to the root form, such as from "flooding" to "flood". For different languages, different lists of stop-words and stemming algorithms have to be applied. However, not all languages are supported by both stop-word lists and stemming algorithms. Therefore, a stop-word list [52], which supported all of the seven languages mentioned above, was used. Subsequently, different stemming algorithms from Natural Language Toolkit (NLTK) library [53] were applied on the sentences in different languages.

Many Twitter bots automatically send messages, such as weather reports, weather forecast or advertisements (examples are shown in Table 1). These messages are regarded as noise information. Most of them normally have similar contents or similar text structure after stemming and removing stop-words. These tweets are often sent repeatedly, which is also a way to automatically detect them: if text messages of one user had similar contents or structures for more than three times, this user was added to a black list, the tweets sent by these users were then filtered out from the input data stream. With this approach, for a collection of geotagged tweets in 30 days (as presented in Section 2), 3.6 million geotagged tweets (from 473,004 users) could be reduced to 2.9 million (from 468,051 users). This means that these 4953 blocked users sent 149.0 tweets on average during 30 days, and this behaved obviously different from ordinary social media users. In the pre-processing steps, we did not normalize the texts or group synonyms.

Table 1. Examples of tweets with similar structure of texts.

No.	Text
1	Wind 13.4 mph NW. Barometer 1023.6 hPa, Rising slowly. Temperature 10.2 °C. Rain today 0.0 mm. Humidity 99%
2	Wind 3 kts NW. Barometer 1025.5 hPa, Rising slowly. Temperature 8.8 °C. Rain today 0.0 mm. Humidity 81%
3	Wind 14.4mph NW. Barometer 1034.1hPa, Rising slowly. Temperature 9.3 °C. Rain today 0.0mm. Forecast Settled fine
4	Wind 2.2 mph NW. Barometer 1032.5 mb, Rising slowly. Temperature 10.9 °C. Rain today 7.2 mm. Humidity 99%

Labeling training data is a typical problem for most of the supervised learning approaches, as large amounts of training data are required. In previous research using machine learning, tweets were manually annotated [13,24]. For instance, in [24] the crowdsourcing service Amazon Mechanical Turk was used to employ annotators for labeling texts. At the end, they could collect 5747 annotated tweets for training their classifiers. Thus, the number of training datasets is limited by the annotation time and budget.

Some recent studies are focusing on training of deep neural network models with noisy labels. The result shows that, the performance of these models is not much affected when small parts of the dataset were not precisely labeled [54]. Aiming at an automatic labelling procedure, we identified that historical weather data are suitable indicators for identifying whether a tweet is relevant to rainfall events, as pluvial floods are directly caused by heavy rainfalls and fast storms. An important data source is Weather Underground [55], a platform that offers Weather API [56] to query historical weather records based on the date and location on city level. In order to automatically label tweets into positive and negative examples, we first filtered them according to pluvial flood relevant keywords and only from these subsets weather data were inquired. Thus, keyword filtering was used to search for potential candidates, which could be subsequently used as a training dataset for the text classification tasks.

The whole procedure is shown in Figure 4. First, the collection of tweets was pre-processed. For the following keyword filtering, a keyword list (as shown in Table 2) which contains the concepts such as "flood", "inundation", "rain" and "storm" in all the seven languages was used. All posts which

contained the keywords were then looked up in the historical weather records via the Weather API. When the weather API reported a rainfall, this tweet was assigned with a positive label. If not, it was labeled as negative. By that, all the potential candidates for text classifier training were automatically labelled based on the weather records.

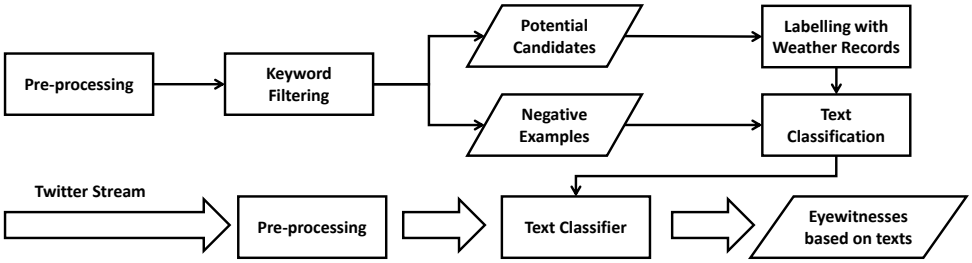

Figure 4. Work flow for training the text classifiers.

Table 2. Keywords used for generating training dataset.

Language	Keywords
English	flood, inundation, deluge, rain, storm
French	inondation, inonder, crue, pluie, orage
German	hochwasser, flut, überschwem, überflut, regen, starkregen, regnen, sturm, unwetter, gewitter
Italian	inondazione, inondare, allagamento, pioggia, diluvio, borrasca, tempestad
Spanish	inundar, inundación, diluvio, aguacero, lluvia, tormenta
Portuguese	inundar, inundação, dilúvio, chuva, chover, tempestade
Dutch	overstroming, zondvloed, stortvloed, regen, storm

From 1 July 2016 to 28 October 2016, about 14.4 million of geotagged tweets were collected within the study area. After filtering, 51,732 tweets (from 36,002 users) were identified as potential training examples. According to Weather API, 36,469 (70.5%) of them were labeled as positive. In order to coarsely verify the automatic labelling, we manually checked 100 randomly selected tweets which were labeled as positive by the weather API: 94 of them are correctly labeled. For such labels with not much noise, text classifiers can be trained.

Training on an imbalanced dataset may lead to over-prediction of the presence of the majority class [57]. For further binary text classifications, a balanced training dataset is required. Therefore, 21,206 randomly selected tweets without any keywords were used as a supplement of the negative training examples to balance the training data. In this way, a balanced dataset was prepared for training the text classifiers. The final training dataset contains totally 72,938 tweets with 65,772 unique words. They were sent by 50,701 users. The average number of words for each document after pre-processing is 6.5.

4.2. Training of Text Classifiers

Following the preparation of a balanced training dataset, text classifiers were trained with five frequently used classical NLP methods, namely naive Bayes [58], random forest [59], SVM with linear kernel [60], SVM with RBF kernel [61] and logistic regression [62]. All these methods are trained based on tf-idf features. As additional method, deep learning using ConvNets for sentence classification was used.

4.2.1. Classical NLP Methods

For the classical NLP methods, the text documents were first transformed into a sparse tf-idf (term frequency - inverse document frequency) [48] matrix, also called the 1-V matrix, where V is the number of unique words in the whole corpus. Term frequency is the raw count of each term in the

sentence. Inverse document frequency indicates the rareness of the words. This value diminishes when the term occurs frequently. The tf-idf matrix can be calculated as follows:

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t \quad (1)$$

$$idf_t = \log \frac{N_d}{df_t} \quad (2)$$

where t stands for term index in the whole corpus, d for document index, N_d for total number of documents and df_t for document frequency of each word. This matrix, was calculated using the methods offered by scikit-learn library [63]. With the normal classification methods in machine learning, the classifiers could be trained based on this tf-idf matrix. Naive Bayes was firstly applied, which is the most basic method for text classification in NLP. It was used as a baseline to demonstrate the performance of the other methods. Random forest, logistic regression, SVM with linear kernel and SVM with RBF kernel are also methods frequently used NLP methods for text classification and the corresponding classifiers were trained separately.

4.2.2. ConvNets for Sentence Classification

Deep learning approaches have achieved an outstanding performance in computer vision, such as ConvNets for image classification [37]. With the development of word embedding techniques, such as word2vec, ConvNets can also be used for sentence classification [49]. Instead of calculating the tf-idf matrix as the training input, each word in the sentences is represented by a word vector using word embedding. These word vectors are learned from the co-occurrence words in a large corpus, for instance, with the word2vec model [50]. Word2vec is a shallow neural network with a single hidden layer. The input is a sparse vector, so called one-hot (1-of-V) vector representing the input word. The output are the probability of the input word to be found nearby the rest of words in the dictionary. These probabilities are calculated using skip-gram. For each input word, skip-gram randomly selects a nearby word to create word pairs and then summarizes the occurrence of the word pairs into probability. The concept *nearby* is defined by a given window size.

After learning with a large corpus, the weights in the hidden layer are the corresponding vector representation for each word. Consequently, the words sharing similar meaning are located close to each other in the vector space. The word embeddings used in this paper were generated based on 20 million unfiltered tweets collected within our study area of Western Europe from 1 July 2016 to 15 December 2016. The total number of vocabulary is 934,063 and the average number of words of each tweets is 6.01. In this case, the python implementations of word2vec in Gensim library (version 0.13.4.1) [64] was used to train this model, which has a default vector dimension set as 300.

ConvNets were then applied on the word embedded sentences with a structure adopted from [49] containing one convolutional layer, one max-pooling layer and one output layer (as illustrated in Figure 5). The output layer has two nodes, which are the topics "relevant" and "irrelevant", respectively. Randomly initialized filters were created with different filter size. After the convolution on the input matrix, feature maps are generated. Then max-pooling is applied on each feature map and a feature vector with the same size as the number of filters is generated. Subsequently, predictions are generated by the soft-max function. The implementation of this ConvNets was based on the Tensorflow [65] (version 1.0.1) framework.

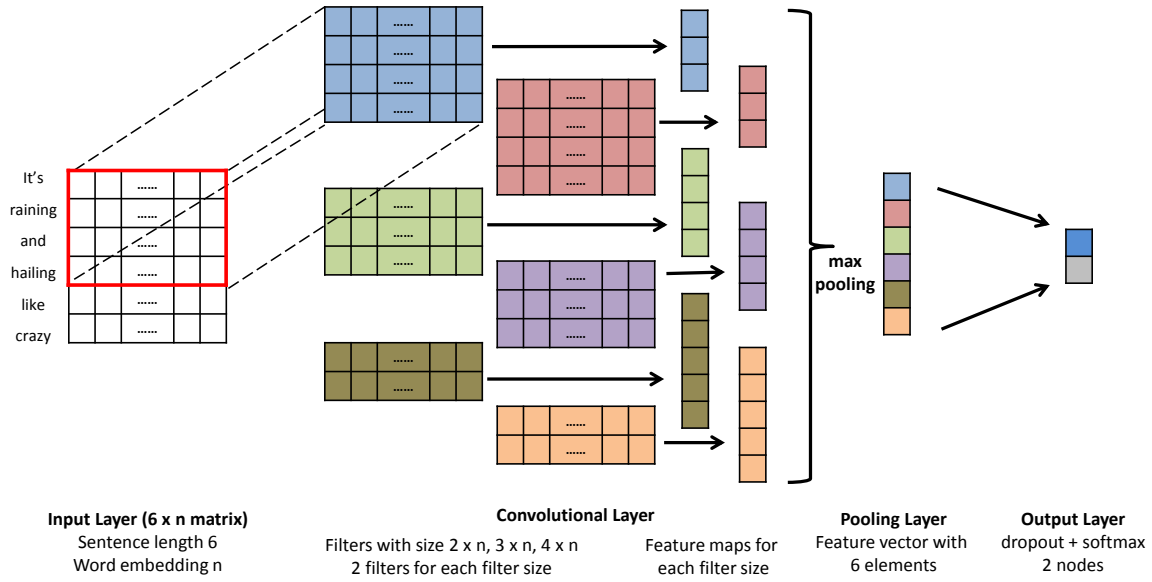

Figure 5. Illustration of ConvNets architecture used for social media text classification (adapted based on Figure 1 in [66]).

4.3. Results and Evaluation

We randomly selected 10% of the dataset (as described in Section 4.1) as test set and used it only for methods comparison. Since most of the classification methods need hyperparameter tuning, we used grid search with 5-fold cross validation on the remaining 90% of dataset to find the optimal hyperparameters for each method (as summarized in Table 3).

Table 3. Parameters used for training the text classifiers.

Method	Parameters
Random Forest	max_depth = 60, n_estimators = 300
Logistic Regression	C = 1.0, penalty = 'l2'
SVM (Linear Kernel)	C = 1.0, gamma = 'auto'
SVM (RBF Kernel)	C = 100.0, gamma = 0.01
ConvNets	learning_rate=0.001

After training the models with the optimal hyperparameters, the performance of all methods are compared and evaluations were given based on the test set with the metrics such as the accuracy, precision, recall and f1-score. F1-score, precision and recall are the metrics calculated based on one single class, the flood and rainfall relevant class. The results are shown in Figure 6 and Table 4. The ROC (Receiver Operating Characteristic) curves (as shown in Figure 7) for each method and area under the curve (AUC) were also calculated and used as criteria for comparing the text classifiers. All experiments in this paper were performed on a PC with Intel Core i7-4790 CPU, 16 GB RAM and one NVIDIA GeForce Titan X GPU. The runtime for training the models is also summarized in Table 4.

As shown in Figures 6 and 7, six text classification methods were compared. The deep learning method using word2vec word embedding and the ConvNet outperformed the other methods and achieved an accuracy of 78.68%. The AUC for ROC of this method is also larger than the others. Except for the naive Bayes, the rest of classical NLP methods using tf-idf matrix as input perform relatively similar. Due to its performance, the trained model using ConvNets was embedded into our application. For the runtime, ConvNets need obvious significantly more time for training. Naive Bayes and logistic regression are the methods which could be trained with less time. Therefore, for an operational use, only the prediction time is relevant, which is similar for all classifiers.

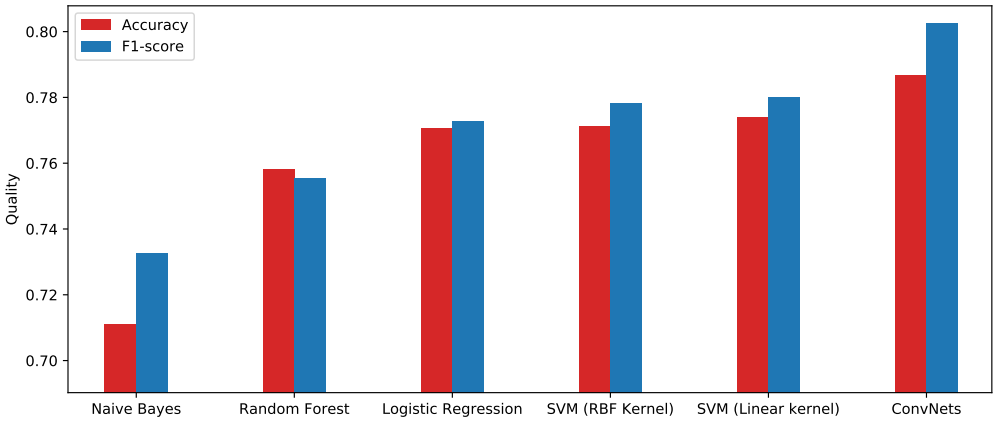

Figure 6. Comparison of text classification methods on test set.

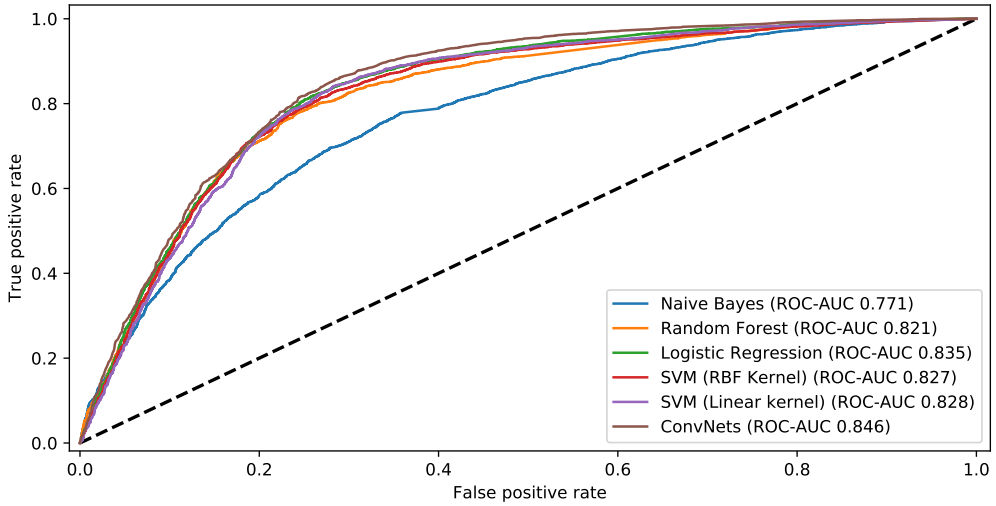

Figure 7. ROC curves of text classification methods.

Table 4. Evaluation of text classification methods.

Method	Accuracy	Precision	Recall	F1-Score	Runtime (s)
Naive Bayes	0.7109	0.6929	0.7769	0.7325	0.02
Random Forest	0.7582	0.7797	0.7324	0.7553	182.1
Logistic Regression	0.7705	0.7793	0.7666	0.7729	0.53
SVM (RBF Kernel)	0.7712	0.7687	0.7881	0.7783	286.0
SVM (Linear Kernel)	0.7739	0.7732	0.7871	0.7801	207.2
ConvNets	0.7868	0.7598	0.8503	0.8025	1124.8

5. Interpretation of Social Media Photos

In this section, we first manually annotated and collected photos for training the classifiers. Transfer learning was then applied to the image data. We systematically tested five classical classification methods for transfer learning, with the aim to find a suitable model to be combined with the real-time pluvial flood detection system.

5.1. Input Training Dataset

The dataset for training the image classifiers has three subsets, which have been collected and labelled by one annotator. Each of them contains 7600 images. Subset 1 contains images which can be frequently seen in social media. They should be irrelevant to flooding or rainfall events. Photos in social

media have their own distribution for each topic, such as artworks, selfies or photos of the surroundings (as shown in Figure 8a); they all have their own proportions in the whole data stream. In order to preserve their distribution, we filtered the tweets with photos from 1 July 2016 to 28 October 2016. The annotator was given randomly selected photos from these tweets, among them 7600 images which are irrelevant to flooding or rainfall events were collected.

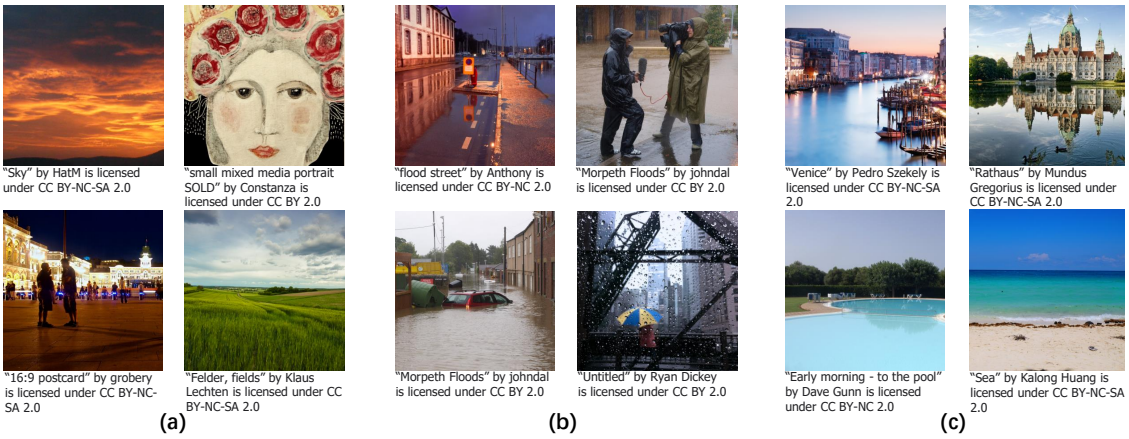

Figure 8. Examples of training dataset: rainfall and flooding irrelevant images (**a**), relevant images (**b**) and images of water surface (**c**).

Since the proportion of flooding and rainfall relevant photos is very small with respect to the whole data stream, it is impractical and time consuming to generate many positive examples by filtering our collected Twitter database. As photos in social media are not limited to device or time, images relevant to flood and rainfall events were manually collected from the Internet using a search engine and search tools provided by Twitter and Instagram. As we concentrated on extracting evidences for flood and rainfall events, Subset 2 included scenarios such as people or vehicles standing beside or in the water, raindrops on the windows or on objects as well as wet or flooded streets (as shown in Figure 8b).

Although in a first test the transfer learned model using these two data sets reached a good performance on the test dataset, it was not capable to make correct predictions when the images contain scenarios including lakes or rivers. Therefore, a second classifier was needed, which makes the classifiers robust against such scenarios. In the end, only the images predicted by both classifiers as positive were regarded as rain and flood relevant images. Therefore, Subset 3 was collected in the same way as the second subset, which contains images of water surfaces, such as rivers, lakes, seaside or swimming pools (as shown in Figure 8c). In this dataset, the flooding and rainfall relevant photos were excluded. It is worth mentioning that the photos in the first subset also contain some photos of the water surfaces, however, the amount of such photos is small and only with respect to the distribution of normal social media images in the data stream.

5.2. Training of Image Classifiers

To interpret whether a user generated photo is relevant to rain and floods or not, we can build a binary image classifier. For training such a model, large amounts of training examples are required, which should contain both positive and negative annotated images. An usual approach to cope with the problem of labelling large amount of training examples is to use transfer learning [41]. The pre-trained ConvNets can serve as a feature generator by removing the output layer. The rest of the weights in the pre-trained model stay unchanged, and the output for each image is then a fixed-size feature vector. As described in the DECAF [39] framework, features could be classified with the classical machine learning such as SVM or logistic regression.

The pre-trained ConvNets utilized in this paper is the GoogLeNet (Inception-V3 model) [67], which was trained based on the ImageNet 2012 Challenge dataset [38]. This dataset contains

1.2 million images categorized into 1000 classes. This pre-trained model is available at the Tensorflow repository [68]. From the description of this model, it could achieve a top-5 error with 4.2% on the test dataset [67]. After removing the output layer, the output for each image is a feature vector with 2048 values. According to the principle of transfer learning, classification was conducted by applying classical machine learning approaches. Since training on an imbalanced dataset may lead to over-prediction of the presence of the majority class [57], the binary classification here was applied on balanced training dataset.

Logistic regression was applied since it is a method frequently used for binary classification. The ensemble methods, such as random forest [59] and gradient boosted trees [69], were also tested. For the three methods above, the implementations in scikit-learn library [63] (version 0.18.1) were used. Furthermore, the xgboost [70] (version 0.6) implementation of the gradient boosted tree was used. Multilayer perceptron with one hidden layer using back propagation was also tested and the implementation was based on the Tensorflow [65] framework.

5.3. Results and Evaluations

Similar to training the text classifiers, 90% of the dataset was used for training. Hyperparameters for each method were tuned by 5-fold cross-validated grid-search. The evaluation was given based on the rest 10%, namely the test set. The hyperparameters used for training the final model for each method are summarized in Table 5.

Table 5. Parameters used for training the image classifiers.

Method	Subset 1 and Subset 2	Subset 2 and Subset 3
Logistic Regression	C = 1000.0 penalty = 'l1'	C = 10000.0 penalty = 'l2'
Random Forest	max_depth = 60 n_estimators = 300	max_depth = 30 n_estimators = 300
Multilayer Perceptron	num_hidden_units = 8 learning_rate = 0.005	num_hidden_units = 8 learning_rate = 0.01
Gradient Boosted Trees	n_estimators = 300 learning_rate = 0.05	n_estimators = 150 learning_rate = 0.1
xgboost	eta = 0.32, gamma = 0.01 max_depth = 15	eta = 0.32, gamma = 0.05 max_depth = 15

The classification methods used for transfer learning were tested firstly on Subset 1 and Subset 2 (as introduced in Section 5.1) and the evaluations are given with accuracy, precision, recall and f1-score on the test set. The ROC curves for each method and AUC were also used as a criteria for evaluation. With the same computer as described in Section 4.3, the training time for each method was also recorded.

As shown in Table 6 and Figure 9, the classifier which was trained based on transfer learning achieved the best performance using the xgboost implementation of gradient boosted trees. Both the accuracy and f1-score reached 92.8% and the AUC of ROC achieved the maximum compared with other methods (as shown in Figure 10). It was followed by the random forest and gradient boosted trees; even the worst case, a simple logistic regression, could also achieve an accuracy of about 88%, which shows the transfer learning approach we applied can really distinguish raining or flooding scenarios in normal daily social media images. When comparing the runtime of each classification method, the gradient boosted trees from scikit-learn is much more time consuming than xgboost; and multilayer perception is the method with the least training time.

Even though high accuracy and high f1-score were achieved on the test dataset, the classifier was still found to be not optimal classifying the images containing water surfaces. Therefore, after the training of the first classifier, a second classifier was trained only to distinguish the topic relevant images from the scenarios containing water surfaces. The same transfer learning approach was utilized

but only with different input data, which contained 7600 images relevant to raining and flooding and another 7600 images containing only images of lakes, rivers or the seaside, as the second and third subsets of dataset shown in Figure 8b,c.

Table 6. Accuracy and f1-score of image classification methods.

Method	Accuracy	Precision	Recall	F1-score	Runtime (s)
Logistic Regression	0.8886	0.9004	0.8752	0.8876	138.8
Multilayer Perceptron	0.8907	0.9745	0.8036	0.8809	22.9
Random Forest	0.9133	0.9497	0.8738	0.9102	117.9
Gradient Boosted Trees	0.9252	0.9342	0.9158	0.9249	669.8
xgboost	0.9295	0.9436	0.9144	0.9288	121.2

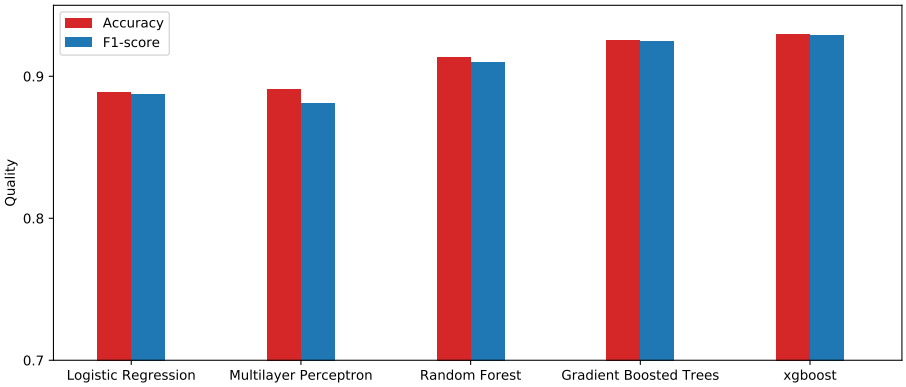

Figure 9. Comparison of image classification methods on test set.

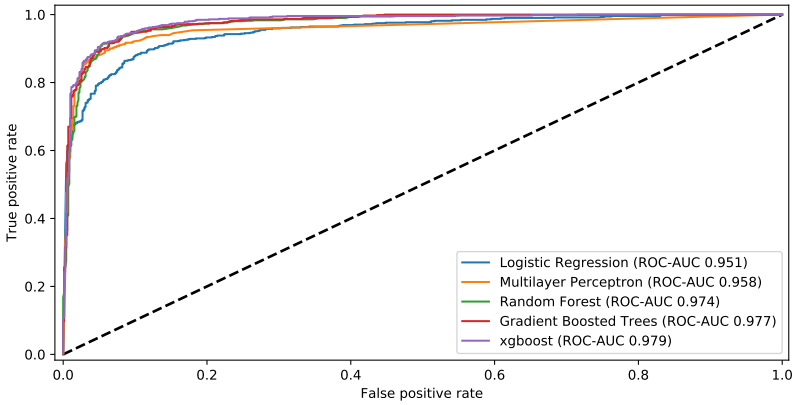

Figure 10. ROC curves of image classification methods.

As shown is Table 7 and Figure 11, similar performance as the first classifier was observed, however, with lower values. Xgboost outperforms the other methods and has a highest accuracy and f1-score. It has also achieved the largest AUC for ROC (as shown in Figure 12). From the result, accuracy of 87.38%, we were convinced that to distinguish the raining or flooding images from images of lake or river is more complicated than from normal social media daily images. Therefore, together with a pre-trained model used as a feature generator, the two trained xgboost models were embedded in our application. Only the images predicted by both classifiers as positive were predicted as rain and flood relevant images.

Table 7. Evaluation of image classification methods.

Method	Accuracy	Precision	Recall	F1-score	Runtime (s)
Logistic Regression	0.8407	0.8453	0.8495	0.8474	221.3
Random Forest	0.8555	0.8763	0.8411	0.8584	158.1
Multilayer Perceptron	0.8625	0.8915	0.8378	0.8638	16.1
Gradient Boosted Trees	0.8695	0.8836	0.8629	0.8731	425.3
xgboost	0.8738	0.8872	0.8679	0.8774	134.2

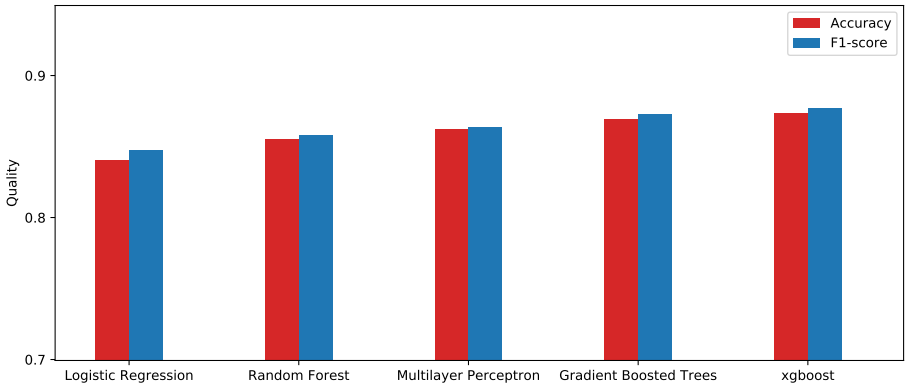

Figure 11. Comparison of image classification methods on test set.

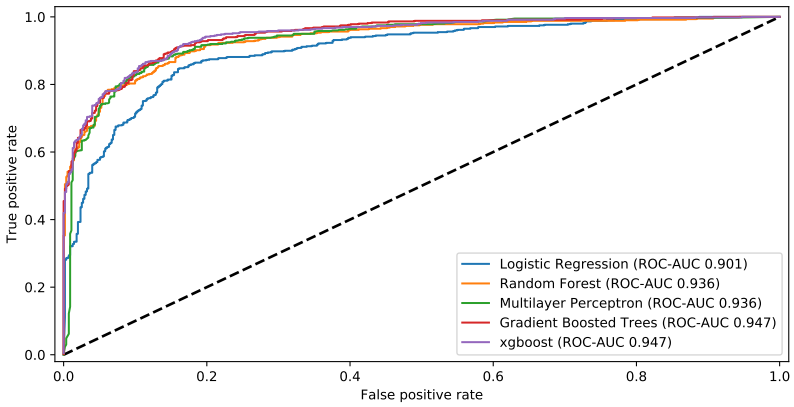

Figure 12. ROC curves of image classification methods.

We conducted a visual inspection of the wrongly classified photos (false positives). They can be generally grouped into three categories. Firstly, many photos with water surfaces in relative dark color were wrongly classified. Secondly, the images containing reflecting area (e.g., windows), which was similar to water reflection were sometimes not well classified. Lastly, photos containing fountains or springs, which have contents like water drops, were also hard to be classified.

6. Detection of Heavy Rainfall and Flooding Events

In this section, only the geotagged tweets containing both texts and images were processed to detect heavy rainfall and flooding events. Only the tweets with positive predictions from both filters were regarded as high quality eyewitnesses for such events. Subsequently, events were detected with spatiotemporal clustering and a hot spot map was generated using Getis-Ord Gi* [71] with respect to the city administrative regions.

For this research, some assumptions are needed. An elementary prerequisite for our approach is the availability of coordinates of the social media posts. They are given by social media platforms and are of heterogeneous quality because of user privacy strategies or device differences. Since improving

user location precision for VGI is not the focus of this research, the posted coordinates are regarded as the location that is related to the contents of the corresponding post. This is also one of the limitations of this work. The contents of some tweets may not be always associated with the posted coordinates. Therefore, spatiotemporal clustering is used for event detection to eliminate this effect. However, for regions strongly connected to each other, improvements are needed for the future works.

Moreover, the main focus of this paper lies on pluvial flood events, thus during the training of classifiers, texts and images containing rainfall relevant information were taken into consideration. However, this does not mean that our models are also designed for distinguishing the pluvial flood events from fluvial or coastal flooding.

6.1. Event Detection with Spatiotemporal Clustering

Spatiotemporal clustering can be used to detect events based on spatiotemporal patterns among data points. ST-DBSCAN [72] is an extension of the density-based clustering method DBSCAN [73] into spatiotemporal space. Three parameters are needed for this method: the maximum spatial distance ε_1, the maximum time difference ε_2 and the minimum number of points to form a cluster $MinPts$. Since users may send several posts at the same place and same time with very similar contents, posts from one single user may already be enough to create the spatiotemporal clusters without confirming from others. Therefore, instead of using the minimum number of tweets, $MinPts$ is redefined in this case as the minimum number of different Twitter users.

For cities of different size, the spatial distribution of Twitter users varies significantly. Since the estimation of such parameters is not the focus of this study, the parameters for spatiotemporal clustering were set to be changeable by the end users. They can adjust them to find the optimal parameter combinations for their own city. According to the literature, the cell size of intense rain is generally less than 10 km in the UK [74] and rain with duration of 3 h contributes the most to total summer precipitation [75]. With this guidance, different combinations were checked and visually compared. At the end, an optimal setting used for the results in London (as shown in Figure 13) is $\varepsilon_1 = 8$ km, $\varepsilon_2 = 1.5$ h and $MinPts = 3$ users.

Figure 13. Spatiotemporal cluster detected by ST-DBSCAN (pluvial flood in London on 26 June 2016, individual tweet are represented as red marker, green markers are the aggregated tweets only for visualization).

6.2. Polygon Based Hot Spot Detection with Getis-Ord Gi*

Getis-Ord Gi* [71] is one of the frequently used geostatistics methods for hot spot detection. This method also takes the local neighbourhood into account. In this case, we used administrative polygon data for the cities to represent the local neighbouring relations. This method was applied to find the statistical hot spots for the extracted rainfall and flood relevant tweets. The principle of Getis-Ord Gi* is to compare local averages to global averages. The results after applying this method are the z-scores, which represent the statistical significance. They indicate the particular value for each polygon relative to the global average. Z-scores are frequently used to determine the confidence threshold. A z-score greater than 1.65 represents for a 90% significance, greater than 1.96 for 95%, greater than 2.58 for 99% and greater than 3.29 for 99.9% [76].

The number of tweets in each part of the city is different because of the difference in social media users' density. A simple hot spot detection directly based on the number of topic-relevant tweets may frequently lead to the appearance of hot spots at the city center or somewhere more people are living. To avoid this situation, we aggregated the total number of tweets in 90 days at the same city and calculated an average number of daily collected tweets for each polygon. An example in Paris, France was generated and shown in Figure 14. The polygons represent the 80 administrative districts provided by Open Data Paris [77]. It is obvious to find that the areas including places of interests or shopping zones in Paris are highlighted. This statistic was used as a basis for inspecting the places where normally few tweets are sent, but suddenly a large number of tweets appears at that area. This may indicate a more reasonable hot spot region.

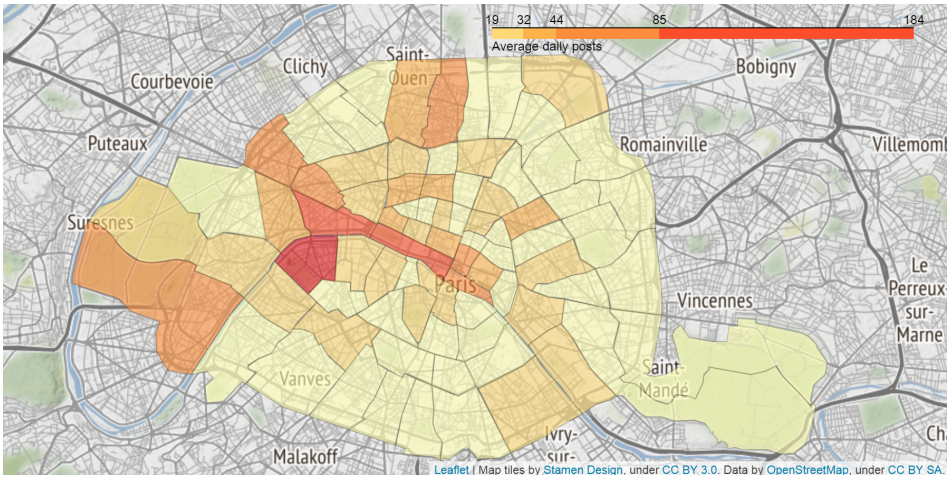

Figure 14. Map of daily average number of tweets based on aggregation of 90 days' tweets.

The coordinates of many tweets are only in city level, for instance, user may provide the single point coordinate representing 'Paris, France' when they sent a social media post. Such imprecise coordinates were also recorded. Thus, these Tweets representing the cities were filtered out before the hot spot detection. After that, the ratios of the number of filtered Tweets and the daily average number of Tweet were calculated for each polygon. Based on the tweets collected in Paris on 3 June 2016, a map of ratios (as shown in Figure 15) was generated. This ratio map is then used as the input for Getis-Ord Gi* hot spot detection. From the result, a map of the z-scores (as shown in Figure 16), a situation in Paris could be identified, showing that the regions along the river bank of the Seine were highlighted during this fluvial (river) flood event. Comparing with Figure 15, it could achieve a better neighbouring consistency.

For the highlighted region as shown in Figure 16, a z score of 3.1 indicates an over 99% confidence, that a cluster of rainfall and flooding relevant tweets existed in that highlighted area. For the point dataset with less than 30 points, the hot spots detection with Getis-Ord Gi* are considered to be not

reliable [76]. Therefore, hot spot detection was only applied, when more than 30 event relevant tweets were extracted by the text and image filters.

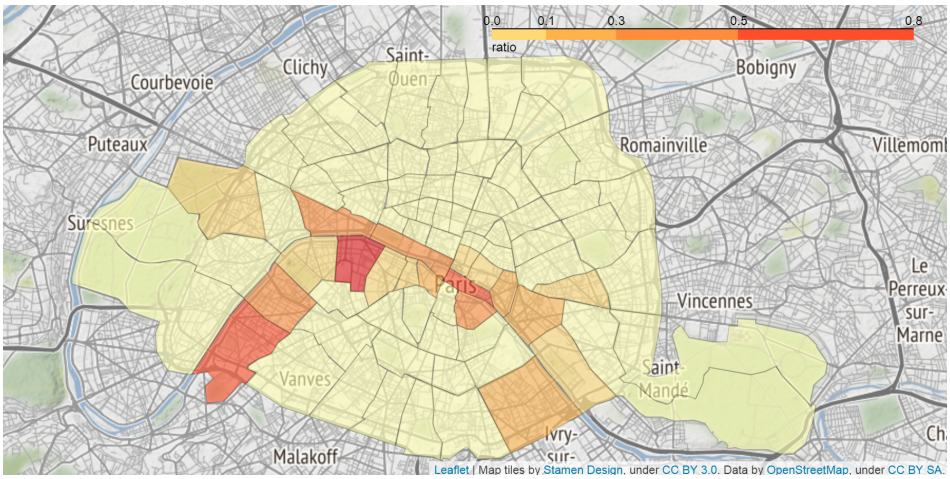

Figure 15. Ratio map on 3 June 2016 in Paris.

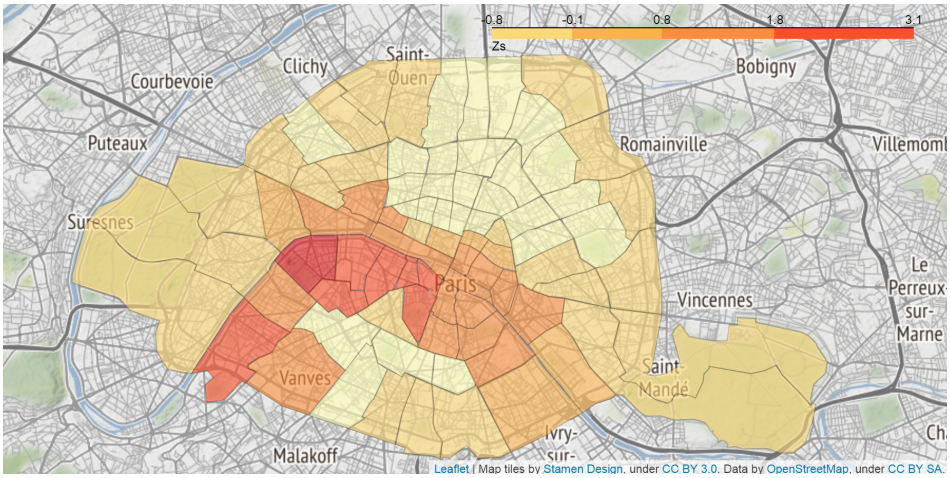

Figure 16. Hot spots detected by Getis-Ord Gi* on 3 June 2016 in Paris.

7. Visualization of the Pluvial Flood Relevant Information

A further test of our framework was applied during the pluvial flood in Berlin, Germany on June 29 2017. A heavy rainfall stroke Berlin and led to severe inundation in the city and failure of the drainage systems [6]. Our application could generate for each day a report with eyewitnesses of the rainfall or flooding events. The eyewitnesses are then visualized as clustered point markers. After clicking the marker clusters, the detailed information of each tweet can be accessed by opening the links in pop-up window at the user given locations. By this approach, overlaps of data points are avoided. Spatiotemporal clusters are visualized as a light blue circle and the radius is set as the bigger eigenvalue calculated based on the data points belong to the same spatiotemporal cluster. Hot spots are also detected based on the prediction from both text and image classifiers and visualized as a choropleth map (as shown in Figure 17). The polygons represent the 138 regions defined by Life-World Oriented Spaces (LOR) [78], which is a partition of the city of Berlin frequently used for statistic and demography. It is also available under Berlin Open Data [79].

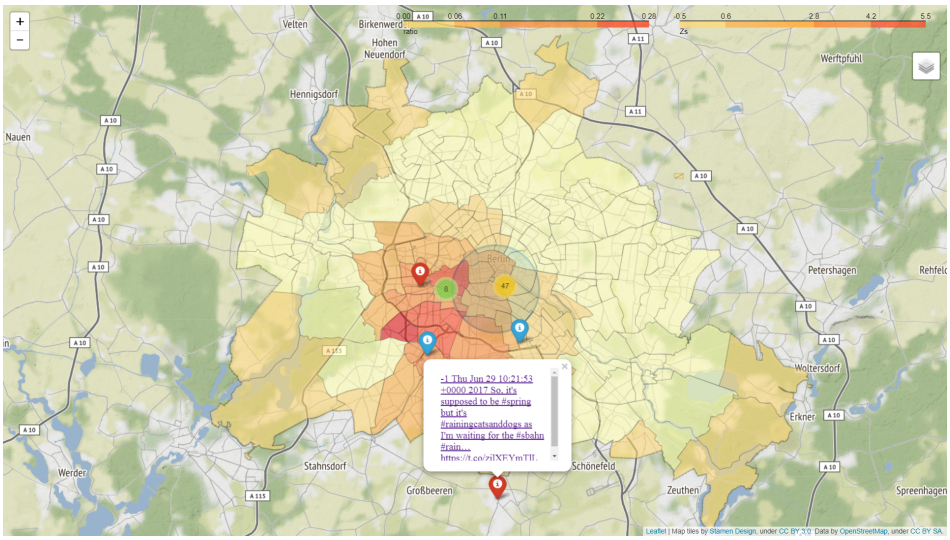

Figure 17. Screen-shots of the web map application (pluvial flood in Berlin on 29 June 2017).

8. Analyses and Comparison with External Data Source

In the previous sections, user-generated texts and photos were used to identify flood and rainfall relevant social media posts. As we aim to evaluate whether the extracted tweets are relevant for the real world events, additional data can be used for a correlation analysis. Since the pluvial floods are normally associated with heavy rainfall events, rainfall intensity can be additional information which is latently related to the occurrence number of flooding relevant tweets. In this case, we accessed the precipitation data recorded by Weather Underground [55]. As classifiers were trained separately for images and texts, three strategies can be compared, namely image-based filtering, text-based filtering and filtering based on both texts and images. Two case studies in Paris and London are given.

For the first case study, correlation analysis is conducted based on the tweets filtered during 45 days from 17 May 2016 to 30 June 2016 in Paris. In this time range, a fluvial flood event has happened. 111,500 geotagged tweets containing both texts and images were collected. After filtering by the text classifier, 2093 tweets are classified as flood relevant. 6431 tweets are classified as flood relevant based on user generated photos. With the confirmation from both text and image classifiers, 690 flooding relevant tweets were extracted. Subsequently, we manually checked these extracted tweets, 616 of them are correctly classified, thus a precision about 89.3% was achieved.

Since each day may have a different numbers of tweets in total, ratios between the topic relevant tweets and total number of tweets on the same day are calculated for the three strategies. As shown in Figure 18, proportions of tweets filtered by the three strategies are presented and the red solid line indicates the precipitation data in millimeter. Correlations between the results from the three strategies and precipitation were calculated and summarized in Table 8. From the results, only a relative small correlation exists between the text-based filtering and the precipitation records, and the other two strategies are almost uncorrelated with the precipitation data. A peak can be identified from the VGI data on 3 June 2016, which is exactly the fluvial flood event on 3 June 2016 [80]. It should be noted, that there was no rain on that day, as indicated by the very low precipitation value. The peak identified by the VGI filter therefore identifies the peak in the fluvial flood and not in the rainfall.

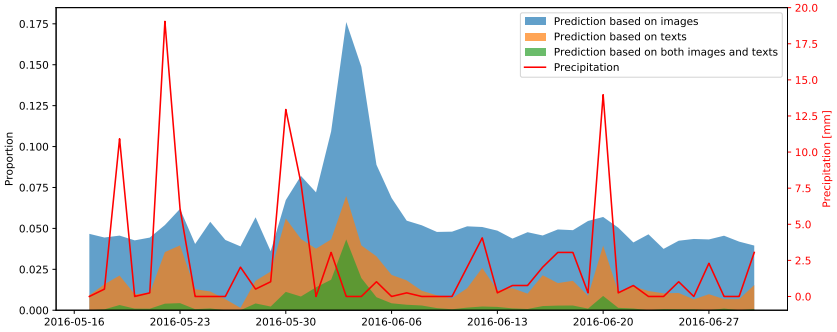

Figure 18. Comparison of the retrieval strategies (Paris, 17 May–30 June 2016).

Table 8. Correlations between the proportion of topic related tweets and rainfall intensity.

Prediction	Correlation	*p*-Value
Prediction based on images	0.0108	0.9439
Prediction based on texts	0.4927	0.0006
Prediction based on both images and texts	0.1063	0.4870

For the second case study, the correlation analysis is conducted based on the tweets filtered from 17 June 2016 to 30 June 2016 in London. As shown in Figure 19 and Table 9, a much stronger correlation can be identified compared to the previous case. On 23 June 2016, a pluvial flood happened in London [81] and the peak on that day can also be identified. In this case, image-based filtering has higher correlation than the others, which shows that the filtering by the image classifier is more sensitive to the real rainfall events.

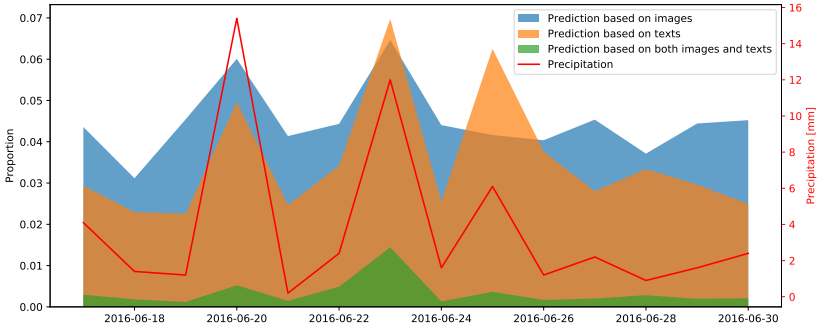

Figure 19. Comparison of the retrieval strategies (London, 17–30 June 2016).

Table 9. Correlations between the proportion of topic related tweets and rainfall intensity (London, 17–30 June 2016).

Prediction	Correlation	*p*-Value
Prediction based on images	0.8360	0.0002
Prediction based on texts	0.7685	0.0013
Prediction based on both images and texts	0.7208	0.0036

In summary, from the two case studies above, we can identify a strong correlation for a time range with pluvial flood, however, when a fluvial flood happens, the correlation becomes weaker. The ground truth used for correlation cannot represent such flooding events properly. Instead of

using precipitation, river gauge can be a potential dataset for calculating such correlation. Therefore, the approach presented in this paper is able to detect pluvial flood events, but not able to distinguish pluvial flood from fluvial and coastal floods.

Furthermore, we also noticed that, pluvial flood events are different from fluvial flood in the sense of spatial distribution of the relevant tweets. As a matter of fact, to be seen clearly in Figure 20 right, a fluvial flood event occurs close to a river, therefore most of the relevant information are accumulated near the river. However, for a pluvial flood event (as shown in Figure 20, left), the extracted tweets distribute much evenly in space. In this way, there is also great potential to distinguish different types of flooding events from the spatial patterns of the extracted social media posts.

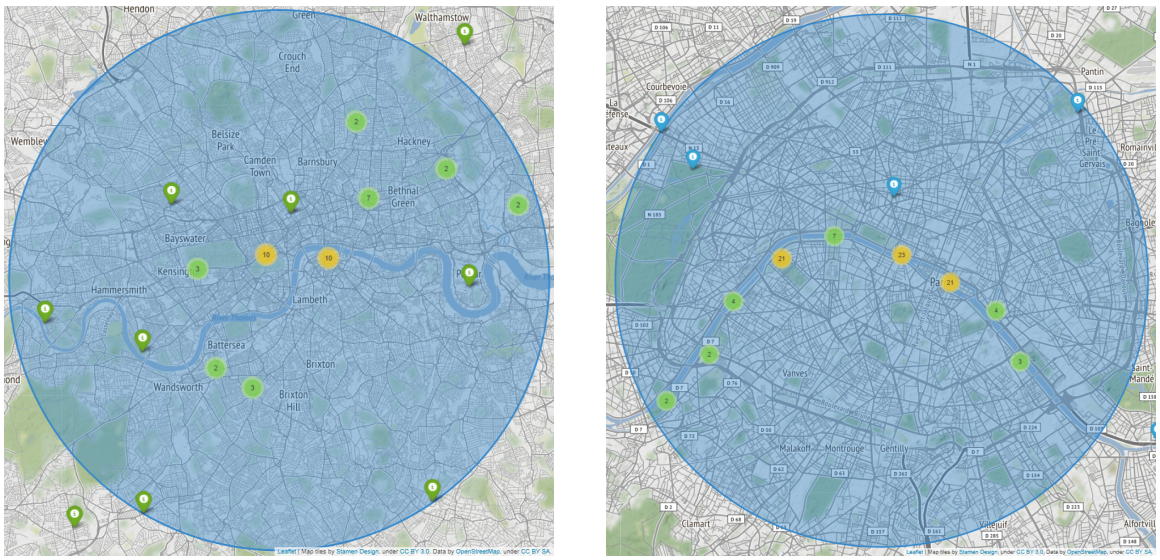

Figure 20. Comparison of pluvial flood event (**Left**) London, 23 June 2016) and fluvial flood event (**Right**) Paris, 3 June 2016).

9. Conclusions

In summary, this article has described a framework to collect, process and analyze pluvial flood relevant information from the social media platform Twitter. The extraction of relevant information takes not only the textual information into consideration, but also user generated photos as supplements to find high quality eyewitnesses for such events. These individual cues for events are subsequently aggregated using spatiotemporal clustering to extract significant clusters in space and time and ignore the outliers. Finally, a document in the form of a map was generated. It visualizes the high quality topic relevant tweets, the spatiotemporal clusters and hot spots of the city for each day. In this paper, we performed the evaluation on the fixed text and image training dataset and filtered the real Twitter stream data. Different filtering strategies are compared with respect to the precipitation data. The case study in London provide evidence that, the extracted number of flood and rainfall relevant tweets are correlated with the precipitation record. The work demonstrated in this paper will be part of a real-time pluvial flood prediction system and serve the city Hanover for emergency response in the future [82]. The prediction will be based on the hydrological simulation. The VGI extracted from social media will be used for real-time event detection and validation of the hydrological models after the pluvial flood events.

Crowdsourcing has a huge potential for many relevant applications. At the same time, however, there are also deficiencies, which should be tackled in future work. A first issue is the inherent uncertainty of the data and also the fact that false information may be uploaded, intentionally or unintentionally. Thus, better methods for detecting fake information are needed, which may improve the information quality. A second issue relates to the location information. Since more and more

social media platforms allow users to send only rough locations in order to protect their privacy, further studies are desirable to improve the localization precision, e.g., based on their daily routes or social network connections.

In this approach, we used a very simple way to combine the results of text and image classification. In the future, we would like to develop a combined strategy which can also make reasonable predictions for the tweets that only have texts or photos, e.g., using a probabilistic approach. Since the training of the classifiers is currently based on a prior given training dataset, an online learning approach which can take the recently collected labelled tweets into consideration is also desirable. For the current text classifier, the model was trained based on a dataset with the mix of documents in seven languages. In the next step, we will train languages specific classifiers to avoid a biased distribution of different languages in the dataset. As the training examples were automatically labelled by weather data, additional annotations are also needed to investigate the effect of label noise. Using neural networks for NLP is currently an active area of research. Many recently proposed approaches are using architectures related to Recurrent Neural Network [83]. We should also test these methods in the future. Moreover, parallel processing architectures, such as Spark Streaming [84], can be applied to make the processing procedure more efficient.

Acknowledgments: The authors would like to acknowledge the support from BMBF funded research project "EVUS — Real-Time Prediction of Pluvial Floods and Induced Water Contamination in Urban Areas" (BMBF, 03G0846A). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a GeForce Titan X GPU used for this research. The publication of this article was funded by the Open Access Fund of the Leibniz Universität Hannover.

Author Contributions: Monika Sester and Yu Feng proposed the original idea of this paper; Yu Feng designed and performed the experiments; Yu Feng wrote the paper; Both authors discussed the results and revised the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUC	Area Under the Curve
ConvNets	Convolutional Neural Networks
LDA	Latent Dirichlet allocation
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
tf-idf	Term Frequency - Inverse Document Frequency
VGI	Volunteered Geographic Information

References

1. Three Common Types of Flood Explained. Available online: http://www.intermap.com/risks-of-hazard-blog/three-common-types-of-flood-explained (accessed on 7 November 2017).
2. Shoothill GaugeMap. Available online: http://www.gaugemap.co.uk/ (accessed on 7 November 2017).
3. NOAA Tides & Currents. Available online: https://tidesandcurrents.noaa.gov/ (accessed on 7 November 2017).
4. Real-Time Prediction of Pluvial Floods and Induced Water Contamination in Urban Areas. Available online: https://www.pluvialfloods.uni-hannover.de/pluvialfloods0.html?&L=1 (accessed on 7 November 2017).
5. Wang, Y.; Wang, T.; Ye, X.; Zhu, J.; Lee, J. Using social media for emergency response and urban sustainability: A case study of the 2012 Beijing rainstorm. *Sustainability* **2016**, *8*, 25, doi:10.3390/su8010025.
6. Zu viel Wasser für Berlin: Stadt Versinkt im Verkehrs-Chaos—B.Z. Berlin. Available online: http://www.bz-berlin.de/berlin/unwetterwarnung-berlin-wetter (accessed on 7 November 2017).
7. Netatmo. Available online: https://www.netatmo.com/ (accessed on 7 November 2017).

8. Google Flue Trend. Available online: https://www.google.org/flutrends/about/ (accessed on 7 November 2017).
9. Ginsberg, J.; Mohebbi, M.; Patel, R.; Brammer, L.; Smolinski, M.; Brilliant, L. Detecting influenza epidemics using search engine query data. *Nature* **2009**, *457*, 1012–1014.
10. Lazer, D.; Kennedy, R.; King, G.; Vespignani, A. The parable of Google Flu: Traps in big data analysis. *Science* **2014**, *343*, 1203–1205.
11. Goodchild, M.F. Citizens as sensors: The world of volunteered geography. *GeoJournal* **2007**, *69*, 211–221.
12. Zook, M.; Graham, M.; Shelton, T.; Gorman, S. Volunteered geographic information and crowdsourcing disaster relief: A case study of the Haitian earthquake. *World Med. Health Policy* **2010**, *2*, 7–33.
13. Sakaki, T.; Okazaki, M.; Matsuo, Y. Earthquake shakes Twitter users: Real-time event detection by social sensors. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; ACM: New York, NY, USA, 2010; pp. 851–860.
14. Earle, P.S.; Bowden, D.C.; Guy, M. Twitter earthquake detection: Earthquake monitoring in a social world. *Ann. Geophys.* **2011**, *54*, 708–715, doi:10.4401/ag-5364.
15. Crooks, A.; Croitoru, A.; Stefanidis, A.; Radzikowski, J. #Earthquake: Twitter as a Distributed Sensor System. *Trans. GIS* **2013**, *17*, 124–147, doi:10.1111/j.1467-9671.2012.01359.x.
16. Herfort, B.; de Albuquerque, J.P.; Schelhorn, S.J.; Zipf, A. Exploring the geographical relations between social media and flood phenomena to improve situational awareness. In *Connecting a Digital Europe through Location and Place*; Huerta, J., Schade, S., Granell, C., Eds.; Springer: Cham, Switzerland, 2014; pp. 55–71.
17. Schnebele, E.; Cervone, G. Improving remote sensing flood assessment using volunteered geographical data. *Nat. Hazards Earth Syst. Sci.* **2013**, *13*, 669–677, doi:10.5194/nhess-13-669-2013.
18. Terpstra, T.; Stronkman, R.; de Vries, A.; Paradies, G.L. Towards a realtime Twitter analysis during crises for operational crisis management. In Proceedings of the 9th International ISCRAM Conference, Vancouver, BC, Canada, 22–25 April 2012; Simon Fraser University: Vancouver, BC, Canada, 2012.
19. De Longueville, B.; Smith, R.S.; Luraschi, G. "OMG, from here, I can see the flames!": A use case of mining Location Based Social Networks to acquire spatiotemporal data on forest fires. In Proceedings of the 2009 International Workshop on Location Based Social Networks—LBSN'09, Seattle, WA, USA, 3 November 2009; ACM: New York, NY, USA, 2009.
20. Wang, Z.; Ye, X.; Tsou, M.H. Spatial, temporal, and content analysis of Twitter for wildfire hazards. *Nat. Hazards* **2016**, *83*, 523–540, doi:10.1007/s11069-016-2329-6.
21. De Longueville, B.; Luraschi, G.; Smits, P.; Peedell, S.; Groeve, T.D. Citizens as sensors for natural hazards: A VGI integration workflow. *Geomatica* **2010**, *64*, 41–59.
22. Fuchs, G.; Andrienko, N.; Andrienko, G.; Bothe, S.; Stange, H. Tracing the German centennial flood in the stream of tweets: first lessons learned. In Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information, Orlando, FL, USA, 5–8 November 2013; ACM: New York, NY, USA, 2013; pp. 31–38.
23. Li, Z.; Wang, C.; Emrich, C.T.; Guo, D. A novel approach to leveraging social media for rapid flood mapping: A case study of the 2015 South Carolina floods. *Cartogr. Geogr. Inf. Sci.* **2017**, *45*, 97–110, doi:10.1080/15230406.2016.1271356.
24. Karimi, S.; Yin, J.; Paris, C. Classifying Microblogs for Disasters. In Proceedings of the 18th Australasian Document Computing Symposium, Brisbane, Australia, 5–6 December 2013; ACM: New York, NY, USA, 2013; pp. 26–33.
25. Feng, Y.; Sester, M. Social media as a rainfall indicator. In Proceedings of the Societal Geo-Innovation: Short Papers, Posters and Poster Abstracts of the 20th AGILE Conference on Geographic Information Science, Wageningen, The Netherlands, 9–12 May 2017; Bregt, A., Sarjakoski, T., van Lammeren, R., Rip, F., Eds.; Wageningen University & Research: Wageningen, The Netherlands, 2017.
26. Fohringer, J.; Dransch, D.; Kreibich, H.; Schröter, K. Social media as an information source for rapid flood inundation mapping. *Nat. Hazards Earth Syst. Sci.* **2015**, *15*, 2725–2738, doi:10.5194/nhess-15-2725-2015.
27. Bischke, B.; Bhardwaj, P.; Gautam, A.; Helber, P.; Borth, D.; Dengel, A. Detection of Flooding Events in Social Multimedia and Satellite Imagery using Deep Neural Networks. In Proceedings of the Working Notes Proceeding MediaEval Workshop, Dublin, Ireland, 13–15 September 2017.

28. Avgerinakis, K.; Moumtzidou, A.; Andreadis, S.; Michail, E.; Gialampoukidis, I.; Vrochidis, S.; Kompatsiaris, I. Visual and textual analysis of social media and satellite images for flood detection@ multimedia satellite task MediaEval 2017. In Proceedings of the Working Notes Proceeding MediaEval Workshop, Dublin, Ireland, 13–15 September 2017.
29. Silvestro, F.; Gabellani, S.; Giannoni, F.; Parodi, A.; Rebora, N.; Rudari, R.; Siccardi, F. A hydrological analysis of the 4 November 2011 event in Genoa. *Nat. Hazards Earth Syst. Sci.* **2012**, *12*, 2743–2752, doi:10.5194/nhess-12-2743-2012.
30. Twitter: Number of Monthly Active Users 2010–2017. Available online: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/ (accessed on 7 November 2017).
31. Twitter-Docs. Available online: https://dev.twitter.com/streaming/overview (accessed on 7 November 2017).
32. Twitter. Rate Limiting. Available online: https://developer.twitter.com/en/docs/basics/rate-limiting (accessed on 7 November 2017).
33. MongoDB for GIANT Ideas. Available online: https://www.mongodb.com/ (accessed on 7 November 2017).
34. Moniruzzaman, A.B.M.; Hossain, S.A. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv* **2013**, arXiv:1307.0191.
35. Dittrich, A.; Lucas, C. Is This Twitter Event a Disaster? In Proceedings of the AGILE'2014 International Conference on Geographic Information Science, Connecting a Digital Europe through Location and Place, Castellón, Spain, 3–6 June 2014.
36. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1995; pp. 255–258.
37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
38. ImageNet. Available online: http://www.image-net.org/ (accessed on 7 November 2017).
39. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, 21–26 June 2014; pp. 647–655.
40. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems 27, Montreal, QC, Canada, 8–13 December 2014; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; MIT Press: Cambridge, MA, USA, 2014; pp. 3320–3328.
41. Goodfellow, I.; Bengio, Y.; Courville, A. Section 15.2—Transfer Learning and Domain Adaptation. In *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; pp. 328–343.
42. Nogueira, K.; Penatti, O.A.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **2017**, *61*, 539–556, doi:10.1016/j.patcog.2016.07.001.
43. Niessner, R.; Schilling, H.; Jutzi, B. Investigations on the potential of Convolutional Neural Networks for vehicle classification based on RGB and Lidar data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 115–123, doi: 10.5194/isprs-annals-IV-1-W1-115-2017.
44. Ammour, N.; Alhichri, H.; Bazi, Y.; Benjdira, B.; Alajlan, N.; Zuair, M. Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sens.* **2017**, *9*, 312.
45. Iannelli, G.C.; Dell'Acqua, F. Extensive Exposure Mapping in Urban Areas through Deep Analysis of Street-Level Pictures for Floor Count Determination. *Urban Sci.* **2017**, *1*, 16.
46. Zamir, A.R.; Hakeem, A.; Van Gool, L.; Shah, M.; Szeliski, R. Introduction to Large-Scale Visual Geo-localization. In *Large-Scale Visual Geo-Localization*; Springer: Berlin, Germnay, 2016; pp. 1–18.
47. Huang, Q.; Xiao, Y. Geographic Situational Awareness: Mining Tweets for Disaster Preparedness, Emergency Response, Impact, and Recovery. *ISPRS Int. J. Geoinf.* **2015**, *4*, 1549–1568, doi:10.3390/ijgi4031549.
48. Manning, C.D.; Raghavan, P.; Schütze, H. Section 6.2—Term frequency and weighting. In *Introduction to Information Retrieval*; Cambridge University Press: New York, NY, USA, 2008; pp. 107–109.
49. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.

50. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*; Curran Associates, Inc.: Red Hook, NY, USA, 2013; pp. 3111–3119.
51. Lin, Z.; Jin, H.; Robinson, B.; Lin, X. Towards an accurate social media disaster event detection system based on deep learning and semantic representation. In Proceedings of the 14th Australasian Data Mining Conference, Canberra, Australia, 6–8 December 2016.
52. Google Code Archive—Stop-Words. Available online: https://code.google.com/archive/p/stop-words/ (accessed on 7 November 2017).
53. Natural Language Toolkit. Available online: http://www.nltk.org/ (accessed on 7 November 2017).
54. Patrini, G.; Rozza, A.: Menon, A.; Nock, R.; Qu, L. Making Neural Networks Robust to Label Noise: A Loss Correction Approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
55. Wunderground—Weather Underground. Available online: https://www.wunderground.com/ (accessed on 7 November 2017).
56. Weather API: Introduction. Available online: https://www.wunderground.com/weather/api/d/docs (accessed on 7 November 2017).
57. Wei, Q.; Dunbrack, R.L., Jr. The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PLoS ONE* **2013**, *8*, e67863.
58. McCallum, A.; Nigam, K. A comparison of event models for naive Bayes text classification. In Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, Madison, WI, USA, 26–30 July 1998; pp. 41–48.
59. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32, doi:10.1023/A:1010933404324.
60. Dumais, S.; Platt, J.; Hecherman, D.; Sahami, M. Inductive Learning Algorithms and Representations for Text Categorization. In Proceedings of the 7th International Conference on Information and Knowledge Management, Washington, DC, USA, 2–7 November 1998; pp. 148–155.
61. Joachims, T. Text categorization with support vector machines: learning with many relevant features. In *Machine Learning: ECML 1998*; Nédellec, C., Rouveirol, C., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1398, pp. 137–142.
62. Genkin, A.; Lewis, D.D.; Madigan, D. Large-scale Bayesian logistic regression for text categorization. *Technometrics* **2007**, *49*, 291–304.
63. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
64. Deep Learning with Word2vec. Available online: https://radimrehurek.com/gensim/models/word2vec.html (accessed on 7 November 2017).
65. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 7 November 2017).
66. Zhang, Y.; Wallace, B. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. Available online: https://arxiv.org/abs/1510.03820 (accessed on 7 November 2017).
67. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
68. Pre-Trained GoogLeNet (Inception-V3). Available online: http://download.tensorflow.org/models/image/imagenet/inception-2015-12-05.tgz (accessed on 7 November 2017).
69. Friedman, J. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232, doi:10.1214/aos/1013203451.
70. Chen, T.; Guestrin, C. Xgboost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794.
71. Ord, J.K.; Getis, A. Local spatial autocorrelation statistics: distributional issues and an application. *Geogr. Anal.* **1995**, *27*, 286–306, doi:10.1111/j.1538-4632.1995.tb00912.x.
72. Birant, D.; Kut, A. ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data Knowl. Eng.* **2007**, *60*, 208–221, doi:10.1145/2534732.2534741.

73. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the International Conference Knowledge Discovery and Data Mining (KKD'96), Portland, OR, 2–4 August 1996; pp. 226–231.
74. Begum, S.; Otung, I.E. Rain cell size distribution inferred from rain gauge and radar data in the UK. *Radio Sci.* **2009**, *44*, doi:10.1029/2008RS003984.
75. Thorp, J.M.; Scott, B.C. Preliminary calculations of average storm duration and seasonal precipitation rates for the northeast sector of the United States. *Atmos. Environ.* **1967**, *16*, 1763–1774.
76. What Is a Z-Score? What Is a *p*-Value?—ArcGIS Pro. Available online: http://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/what-is-a-z-score-what-is-a-p-value.htm (accessed on 7 November 2017).
77. Open Data Paris—Quartiers Administratifs. Available online: https://opendata.paris.fr/explore/dataset/quartier_paris/information/ (accessed on 7 November 2017).
78. Lebensweltlich Orientierte Räume (LOR) in Berlin. Available online: http://www.stadtentwicklung.berlin.de/planen/basisdaten_stadtentwicklung/lor/ (accessed on 7 November 2017).
79. Geometrien der LOR-Bezirksregionen Berlins—Offene Daten Berlin. Available online: https://daten.berlin.de/datensaetze/geometrien-der-lor-bezirksregionen-berlins-stand-072012 (accessed on 7 November 2017).
80. Paris Floods: Seine at 30-Year High as Galleries Close—BBC News. Available online: http://www.bbc.com/news/world-europe-36446635 (accessed on 7 November 2017).
81. Flash Flooding Causes Chaos in Parts of England—BBC News. Available online: http://www.bbc.com/news/uk-england-london-36471889 (accessed on 7 November 2017).
82. Fuchs, L.; Graf, T.; Haberlandt, U.; Kreibich, H.; Neuweiler, I.; Sester, M.; Berkhahn, S.; Feng, Y.; Peche, A.; Rözer, V.; et al. Real-Time Prediction of Pluvial Floods and Induced Water Contamination. In Proceedings of the 17th International Conference on Urban Drainage, Prague, Czech Republic, 10–15 September 2017.
83. Goldberg, Y. CHAPTER 14: Recurrent Neural Networks: Modeling Sequences and Stacks. In *Neural Network Methods for Natural Language Processing*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2017; pp. 163–176.
84. Spark Streaming. Available online: http://spark.apache.org/streaming/ (accessed on 7 November 2017).