Damage Identification in Social Media Posts using Multimodal Deep Learning

Hussein Mouzannar **Yara Rizk** **Mariette Awad**

Department of Electrical and Computer Engineering
American University of Beirut
{hmm46, yar01, mariette.awad}@aub.edu.lb

ABSTRACT

Social media has recently become a digital lifeline used to relay information and locate survivors in disaster situations. Currently, officials and volunteers scour social media for any valuable information; however, this approach is implausible as millions of posts are shared by the minute. Our goal is to automate actionable information extraction from social media posts to efficiently direct relief resources. Identifying damage and human casualties allows first responders to efficiently allocate resources and save as many lives as possible. Since social media posts contain text, images and videos, we propose a multimodal deep learning framework to identify damage related information. This framework combines multiple pretrained unimodal convolutional neural networks that extract features from raw text and images independently, before a final classifier labels the posts based on both modalities. Experiments on a home-grown database of labeled social media posts showed promising results and validated the merits of the proposed approach.

Keywords

Humanitarian computing, deep neural networks, multimodal learning, natural language processing, visual object recognition.

INTRODUCTION

In times of crisis, it is crucial that emergency responders reach all those affected in a timely manner. However, this task can prove to be challenging when first responders have limited knowledge about survivors; regular emergency communication channels, when accessible by those affected, cannot always handle all calls and prioritize response. With over 2.46 billion social media users (as of 2017) posting thousands of messages during natural disasters (Techradar, 2013), monitoring social media outlets for relevant content has recently become common practice among emergency responders. During Hurricane Sandy, about 25 first aid personnel monitored 2.5 million posts related to Sandy (Baer, 2012), and during Hurricane Harvey, social media came to the rescue of a woman tweeting for help as 911 wasn't responding (Rhodan, 2017). However, the huge number of posts makes it impossible for emergency responders to manually mine posts for relevant information, and takes valuable man-power away from other important tasks. Therefore, automating the extraction of actionable information from social media is essential to fully leveraging the benefits of this abundance of data.

An emerging goal of the humanitarian computing community is to build automated systems that detect and flag social media posts for disaster and crisis data. Such systems can be decomposed into three main modules: (1) retrieving possibly relevant social media posts which entails filtering through huge numbers of irrelevant posts while checking for authenticity and recency among other criteria; (2) classifying filtered posts into multiple damage and disaster categories while checking for informative content; and (3) summarizing posts containing disaster information that would be communicated to emergency responders.

In this work, we focus on module (2), identifying damage to infrastructure and environmental elements, in addition to human casualties in posts. Determining whether severe infrastructure damage has occurred in a highly dense city vs. mild damage to a rural town can aid first responders in recognizing each location's needs and optimally allocating resources; a highly dense city may need more shelter aid than a rural town that may need access to clean water instead. Both locations may have been affected by the same natural disaster or event that caused the damage; in our application, information about the type and degree of damage is more important than the cause of the damage. Identifying human casualties would also allow first responders to provide targeted aid in a timely fashion, possibly reducing the number of fatalities.

Our aim is to build a classifier that can handle multimodal data, such as images and text. One approach is to train an artificial neural network (ANN) on multimodal data, but this approach would not benefit from well-established unimodal models from the literature. Instead, we propose a multimodal framework based on pretrained unimodal models that are then combined to obtain a final classifier. Specifically, we adopt the Inception convolutional neural network (CNN) model (Szegedy et al., 2015), pretrained on ImageNet to process images, and we use a pretrained word embedding model to process the text which are then fed to a single layer CNN. Feature fusion (FF) combines the internal representations generated from the various layers of the CNNs to train a multimodal classifier. On the other hand, decision fusion (DF) combines the predictions of the unimodal classifiers to obtain better predictions by either using decision rules or training a classifier. We evaluate our framework on a home-grown labeled dataset of multimodal social media posts and show the advantages of using both modalities to make decisions.

Next, we present existing work on humanitarian computing, deep learning and multimodal learning. Then, we discuss our proposed methodology, describe the home-grown dataset and report on the experimental results, before concluding with final remarks.

RELATED WORK

Humanitarian Computing

Work leveraging social media posts has been developed for various humanitarian computing applications including event detection (Sakaki et al., 2013), alert systems (Breen et al., 2016), map generation (Cresci et al., 2015a) and actionable information extraction (Ashktorab et al., 2014; Caragea et al., 2011). Identifying the type and degree of damage, whether to buildings, natural landscapes or people, is crucial to extract actionable information and allow first responders to properly allocate resources. Damage identification work can be divided into those that processed text, images or both.

Imran et al. (2014) classified tweets by processing the text only to extract damage related information. Similarly, Cresci et al. (2015b) extracted damage information from tweets using natural language processing techniques to understand the linguistic content of these tweets. Nguyen et al. (2016) adopted deep ANN to process the contents of tweets and classify them based on their relevance.

Focusing on images only, Alam et al. (2017) developed an end-to-end framework to collect, process and assess the type and severity of damage in images using a deep learning approach. Nguyen et al. (2017) fine-tuned CNNs with domain specific images to improve damage identification from visual cues.

Finally, some work has investigated combining both text and images in a multimodal framework. Specifically, Jomaa et al. (2016) determined whether damage in tweeted images was to buildings or natural landscapes by extracting cheap visual features combined with bag-of-word processing of the accompanying text, which outperformed other unimodal classifiers.

Deep Learning

Visual scene understanding and semantic understanding are crucial components of identifying and assessing damage for actionable information extraction. Deep learning is one of the leading approaches in the literature to achieve scene and semantic understanding from large amounts of possibly unlabeled and unstructured data with minimal feature engineering (LeCun et al. 2015).

CNN is one such approach that was first developed for visual scene understanding (Krizhevsky et al., 2012) but has also been successfully applied to text (Kim, 2014) and audio processing (Abdel-Hamid et al., 2014). Trained on large amounts of unlabeled and labeled data, CNN can identify a wide range of object categories with super-human accuracies (Krizhevsky et al., 2012), by transforming the raw input (image pixel values) through successive convolution layers and a nonlinear pooling layer (LeCun et al., 2015). For example, CNN achieved an error rate of approximately 15% on the ImageNet database which contained a million images divided into 1000 categories, almost 11% higher than other methods (Krizhevsky et al., 2012). Inception, a 15-layer CNN, also achieved state-of-the-art results on image recognition tasks (Szegedy et al., 2017). Recurrent neural networks (RNN), another deep learning approach, has performed well on visual scene understanding. Specifically, long short-term memory, a type of RNN, achieved a high accuracy on image caption generation, which requires scene understanding (Xu et al., 2015).

Focusing on semantic understanding, deep learning has been used for word embeddings, where words are represented by dense continuous vectors. Sentence and text understanding using deep learning became possible after the introduction of word embedding tools such as Word2Vec (Mikolov et al., 2013a), which allowed ANN to process the words through these vector representations. State-of-the-art results have been achieved on sentiment analysis (Glorot et al., 2011), sentence classification (Lai et al., 2015), text generation (Vinyals et al., 2015) and other tasks, using deep learning. Furthermore, CNN have contributed to deep learning's success on text classification (Conneau et al., 2016; Kim 2014; Zhang el al., 2015), sentiment analysis (Dos Santos et al., 2014), and learning semantic representations (Shen et al., 2014).

Multimodal Learning

With the diverse sources and properties of data, machine learning algorithms that can combine these various perspectives are necessary to maximize the utility of this data. Multimodal learning deals with this issue by developing or adapting machine learning algorithms to learn from multiple modalities. Guillaumin et al. (2010) improved image categorization by including tags in the feature vectors used to train a multiple kernel learning classifier, which resulted in almost 10% improved accuracy on some classes. Alqhtani et al. (2015) detected the occurrence of various events from Twitter by extracting semantic and visual features to train a k-nearest neighbor (KNN) classifier and achieved up to 8% improvement in classification accuracy compared to unimodal approaches. Jomaa et al. (2016) also adopted a FF approach, where feature vectors contained visual and semantic features extracted from images and text, respectively. These features trained a support vector machine (SVM) classifier that improved damage related classification by approximately 4%. Poria et al. (2016) performed sentiment analysis by processing text, audio and images using FF and DF approaches that exceeded the accuracy of state-of-the-art approaches by at least 20%.

While these references performed feature extraction and supervised learning separately, deep learning multimodal algorithms have been proposed to process raw data in unsupervised learning frameworks. These works include Ngiam et al. (2011) who trained deep ANN to learn representations based on raw audio and visual inputs, and Srivastava et al. (2012) who trained deep Boltzmann machines using images and text.

APPROACHES AND MODELS

The proposed deep learning framework consists of three main components: text processing, image processing and multimodal fusion, which can be at the feature or decision level. Next, we describe the unimodal deep learning classifiers for text and images, before discussing the DF and FF classifiers.

Text classification model

To classify text, we employ a shallow CNN based on Kim's proposed architecture (Kim, 2014). While RNN are popular in natural language processing since they can model the sequential structure of text, CNN have achieved equal if not superior performance in classification tasks (Yin et al., 2017). Proposed models in the literature range from one layer (Kim, 2014) to six layers (Zhang et al., 2015) and even 29 layers (Conneau et al., 2016). Recent empirical evidence has shown that deeper models do not necessarily increase accuracy; a wide and shallow CNN has beaten other deep CNNs on multiple datasets (Le et al., 2017).

The input to our CNN is a matrix where each row is a real-valued vector representation of each word in the caption or tweet. Our word-vectors are obtained from a pretrained word embedding model, a common method when a dataset is small (Collobert et al., 2011; Iyyer et al., 2014; Socher et al., 2011). The model architecture used to learn an internal representation of the text is depicted in Figure 1. The CNN is divided into 3 modules: convolutional layer, max-pooling layer and a fully connected layer with softmax outputs.

1) The convolutional layer is composed of three types of filters with height $H_f = [3,4,5]$, all of width $W = 200$ with n such filters of each type, where the filter weights $w_{i,j}$ are learned. We perform a dot product between the filter and the input matrix $\boldsymbol{x}$, recording the result, and then sliding the matrix one row at a time and repeating the process until the last row is reached. This process produces a vector, $\boldsymbol{c_i} = f(\boldsymbol{w} * \boldsymbol{x} + \boldsymbol{b})$, for each filter i, where $f(x) = max(0, x)$ represents the non-linear activation function called rectified linear unit or ReLU, $*$ is the convolution operation described in the preceding paragraph, and $\boldsymbol{b}$ is a bias term.

2) The max-pooling layer takes $\boldsymbol{c_i}$ and produces $\mathrm{m_i} = \max_{1 \leq k \leq |\mathrm{c_i}|} \mathbf{c_{i_k}}$. The results of each max-pooling operation are concatenated to produce a vector $\boldsymbol{m} = [m_1, m_2, \ldots, m_{3n}]$.

3) The vector $\boldsymbol{m}$ is then fed into a fully connected layer, with dropout to avoid overfitting (Srivastava et al., 2014), followed by softmax outputs for each class.

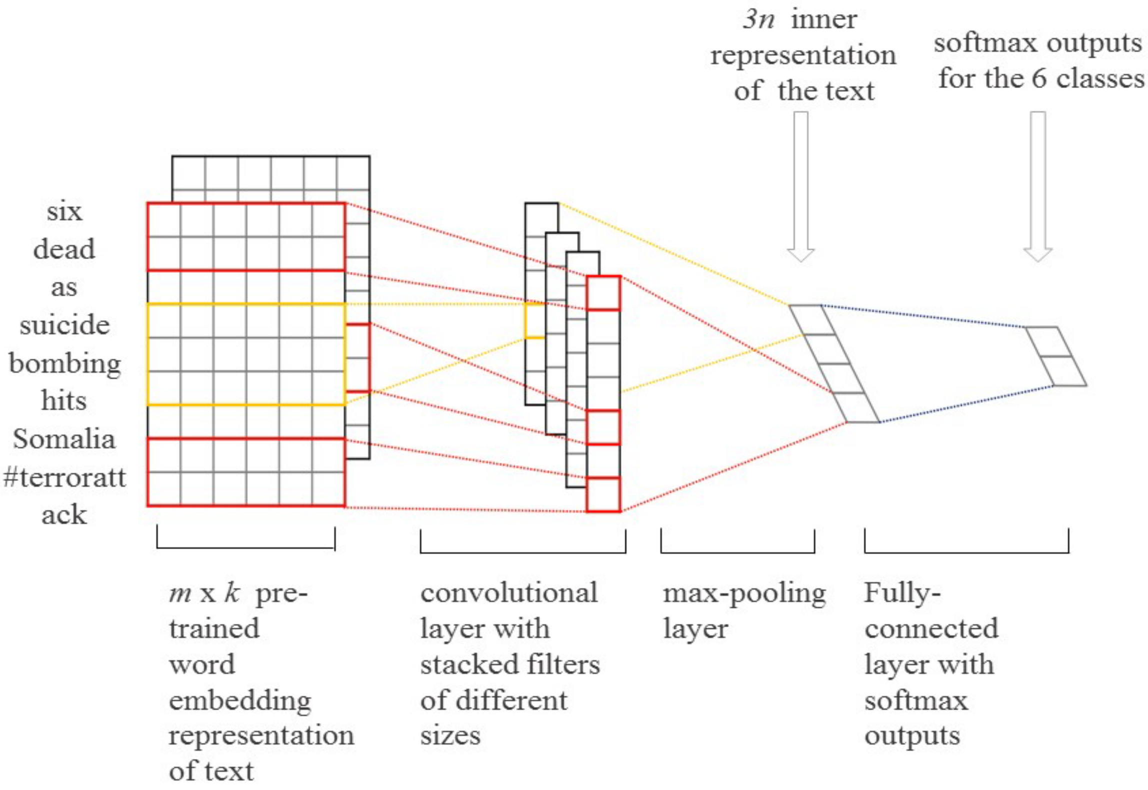

$3n$ inner representation of the text
softmax outputs for the 6 classes
six
dead
as
suicide
bombing
hits
Somalia
#terroratt
ack
m x k pre-trained word embedding representation of text
convolutional layer with stacked filters of different sizes
max-pooling layer
Fully-connected layer with softmax outputs

Figure 1. CNN for Text Classification (adapted from (Kim, 2014))

To reach an optimal performance, we modified the network design and tuned the parameters by following the guidelines in (Zhang and Wallace, 2015), as reported in the experiments section. Furthermore, we compared the GloVe model pretrained on the Twitter dataset of two billion tweets each of dimension 200 (Pennington et al., 2014) to the Word2Vec model trained on a Google News corpus of 3 billion words (Mikolov et al., 2013b), while varying the number of filters and the filter region sizes.

Image classification model

One of the more popular image classification approaches today involves using a CNN pretrained on a large dataset, and then adapting (fine-tuning) the model to the classification task at hand using context or domain specific data (Donahue et al., 2014; Sharif Razavian et al., 2014). The ImageNet dataset (Deng et al., 2009), which consists of over 14 million unlabeled images divided into over 1000 categories, has been commonly used to pretrain deep networks. One explanation for ImageNet's success is the broad range of categories it covers that are applicable to most classification tasks, but no definitive answer has been reached yet (Huh et al., 2016).

Current CNN architectures for image classification tasks include ResNet (He et al., 2016), VGG (Simonyan and Zisserman, 2014) and NASNet (Zoph et al., 2017). We opted to use the Inception architecture, first introduced in 2015 (Szegedy et al., 2015), for its near state-of-the-art accuracies on ImageNet while using a smaller network compared to other models. The network consists of stacked "Inception Modules", as described in Figure 2. Each module combines multiple convolution filters of varying sizes and pooling units. The input to the module, which consists of raw pixel values for the first layer, is concurrently passed through multiple convolution filters of different sizes starting with 1x1 convolution which average out the different channel components. The output of all these filters are concatenated to form the input to the next layer.

Inception-v4, the current Inception architecture shown in Figure 3, is a network of stacked inception modules followed by a softmax layer. The output of the last layer before the softmax block, called the bottleneck feature, is the feature vector used as a representation of the original image for the multimodal model.

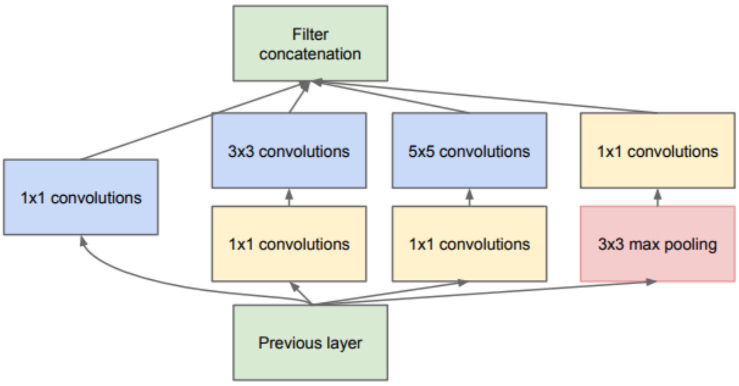


Figure 2. Inception Module Example (Adapted from Szegedy et al., 2015)

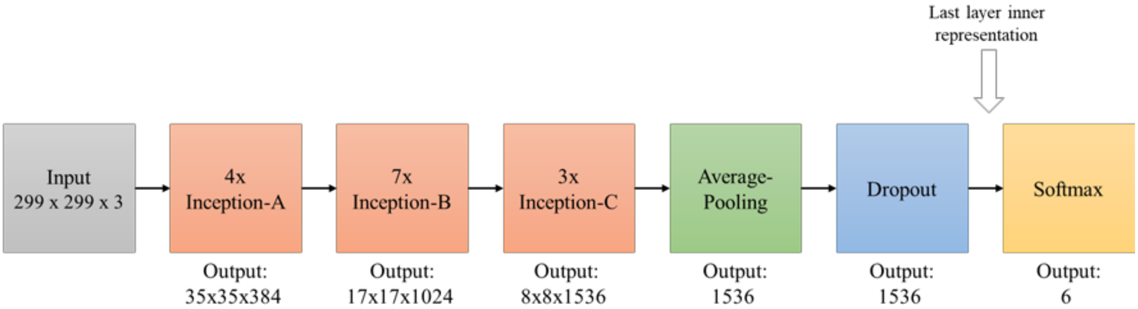


Figure 3. Simplified Inception-v4 Architecture

Decision fusion

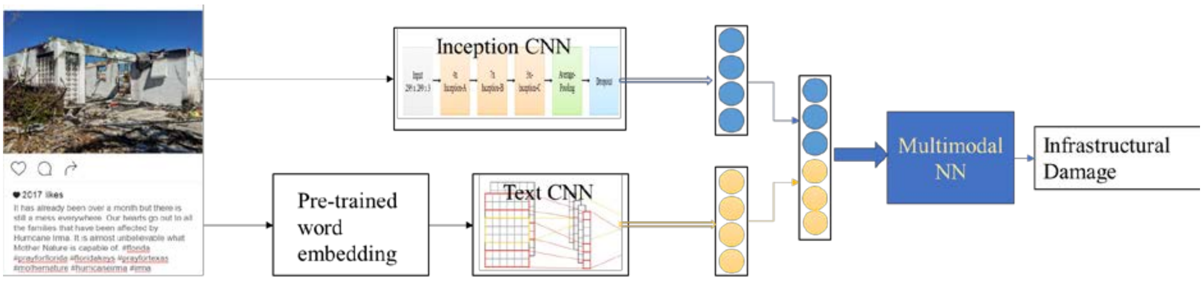
To improve the overall classification accuracy, we turn to approaches in ensemble learning for inspiration on how to combine both modalities, where different learning algorithms are combined to form a better classifier on a given task (Zhou, 2009). The text and image models, $\boldsymbol{h_T}$ and $\boldsymbol{h_I}$ respectively, produce six-dimensional softmax outputs for each input, where the decision is based on the class corresponding to the index with the maximum value. A priori, we believe that both models are equally good, thus, we sum their outputs to obtain a new output vector: $\boldsymbol{h} = \boldsymbol{h_T} + \boldsymbol{h_I}$ and then decide on the class with the highest value; we refer to this approach as maximum decision rule.

It is not usually possible to interpret ANN softmax outputs as uncertainties as much as we would the outputs of classical algorithms such as logistic regression or naive Bayes. We observe that softmax outputs in ANN tend to have very sharp distributions, i.e. approximately equal to 1 for predicted class and 0 for the other classes, the networks were trained using a cross-entropy loss function. Therefore, we apply a softmax function with temperature τ to $\boldsymbol{h_T}$ and $\boldsymbol{h_I}$, to soften their distributions before combining them, as shown in (1). A higher τ implies a softer distribution, i.e. as $\tau \rightarrow \infty$, $\tilde{h}_T(\mathrm{i}) = \frac{1}{6}$.

$$\tilde{h}_T(i) = \frac{exp(h_T(i)/\tau)}{\sum_{j=1}^{6} exp(h_T(j)/\tau)} \forall i \tag{1}$$

However, this approach neglects the accuracy of each model, i.e. if the image model is significantly more informative than the text model, its predictions should be given a higher priority. To remedy this issue, we split our training set into two parts: one for training the models and another, called validation set, to obtain a per-class accuracy estimate for each model: $\boldsymbol{a_T}$ and $\boldsymbol{a_I}$. The class-model accuracy estimates are simply per-class misclassification rates. We scale the modified softmax outputs $\tilde{h}_T$and $\tilde{h}_I$ by their accuracy estimates, then take their sum, $h = a_T^T \tilde{h}_T + a_I^T \tilde{h}_I$, with the final prediction being the class with the maximum weight. This approach is referred to as weighted maximum decision rule.

Another approach is to learn this decision rule, instead of simply using the class accuracies and a maximum decision rule. We can train a meta-classifier on the validation set, where the input is the concatenation of $\boldsymbol{h_T}$ and $\boldsymbol{h_I}$. This approach is called stacking (Wolpert, 1992) in ensemble learning. We experiment using different learning algorithms for our meta-classifier such as linear models, SVM and ANN.



2017 likes

It has already been over a month but there is still a mess everywhere. Our hearts go out to all the families that have been affected by Hurricane Irma. It is almost unbelievable what Mother Nature is capable of. #florida #prayforflorida #floridakeys #prayfortexas #mothernature #hurricaneirma #irma

Figure 4. FF Model

Feature fusion

With DF, only the predictions or final output of the networks are used. Instead of taking the output of the softmax layer of the network, we can take of the outputs of any arbitrary layer of our networks and use them as input to our meta-classifier. Recall that the input to both models is the raw data; no prior feature engineering was performed. The text model uses word embeddings as its first representation of the text and then convolutions are applied on the embeddings to produce a second internal representation of the text. On the other hand, the raw image passes through different layers of convolutions; at each stage, an internal representation of the image is computed. ANN learn a representation of their input with each successive layer (Bengio et al., 2013); these representations can be used as a learned feature of the original data.

The question now is from which layer should we take our features from. In fact, the internal representation at each stage loses some of the information contained in the representation in the previous stage, a fact illustrated by the data processing inequality (Cover and Thomas, 2006), which states that processing will never result in a gain of information. In some sense, it is a sort of lossy compression of the information found in the input. However, what we gain from each stage of processing is a more separable dataset which is easier to classify. This can be seen if we look at the layers in reverse order: the softmax layer only requires a maximum decision rule, the pre-softmax layer requires a single-layer perceptron to achieve good results and so on. Therefore, there exists a trade-off between the ease of classification and the information content in the representations of the network layers. We propose to use the last layer before the softmax layer as our features for the text and the images from their respective models, since they would be easier to classify than earlier layers but still retain more information than the softmax outputs.

As with DF, we are faced with the issue of giving more weight to the more informative modality. Two factors affect the influence of each modality on the overall classification; (1) the dimension of the features: if the text has a higher dimension feature vector than the image, the meta-classifier may give more weight to the text and vice-versa; and (2) the sparsity of the feature representations (contain a large number zeros): a sparse feature vector would contribute less to the decision.

Figure 4 summarizes the FF approach. The images and text CNNs are first trained independently on the training set. Then, a meta-classifier whose input is the concatenation of the features from the layer before last of the networks, is trained on the same training set.

DATA COLLECTION

Next, we describe the collected dataset, and the techniques employed to filter and organize its content.

Captioned image retrieval

Social media platforms such as Instagram, Twitter and Facebook are among the first to report cases of emergencies and crisis situations; they represent a critical source of information. However, these sites are usually overflooded with posts that are either irrelevant or non-informative and make it hard to disseminate actionable information to first responders (Olteanu et al., 2015). Disaster related content filtering has been performed on Twitter posts for both the tweets (Abel et al., 2012; Chowdhury et al., 2013; Jomaa et al. 2016; Olteanu et al., 2015) and images attached to them (Alam et al., 2017; Jomaa et al. 2016). This paper investigates the interplay between the images and the captions in crisis situations. Thus, the Instagram platform represents a more natural choice since posts on Instagram are captioned images, reducing the need to search for tweets with images.

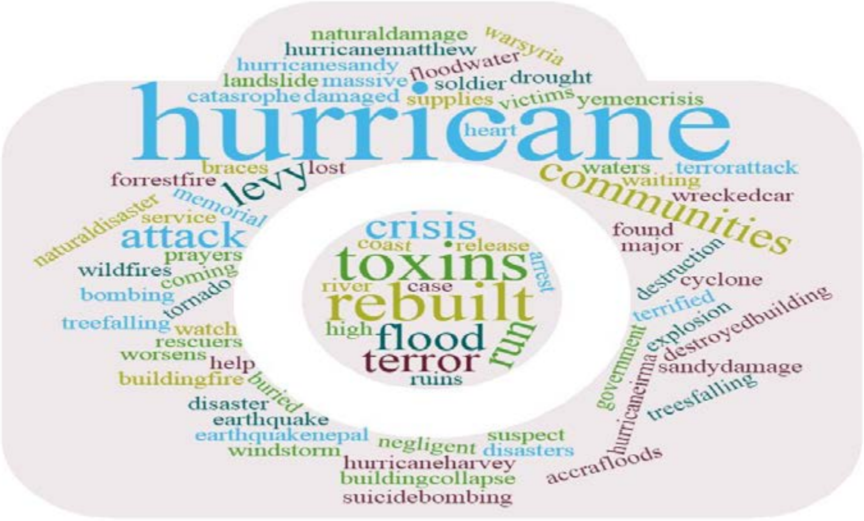


Figure 5. Sample of Hashtags Used for Data Collection

The main method of collecting data from these platforms is through hashtag or keyword queries using their application programming interfaces (API). However, this method heavily relies on the correct choice of hashtags which shape the resultant dataset (Olteanu et al., 2016). Even though hashtags can be considered as a form of tagging, their use is often inconsistent and may result in missing relevant data that was not properly tagged (Potts et al., 2011; Olteanu et al., 2016).

Olteanu et al. (2014) provided a publicly available crisis lexicon, "CrisisLex", which attempted to provide a more representative sample of crisis situations posted on social media. We used their lexicon to collect Instagram and Twitter posts and extended it to include event specific keywords such as "Hurricane Irma", "Hurricane Nate", and "Iran-Iraq Earthquake", to retrieve data from recent disasters. The approach of using a specified lexicon is not suitable for building an automated disaster detection system, as using event specific terms will only be possible during or after the event occurs, and broad terms won't capture the entirety of posts related to a specific event. Therefore, having a dynamic and adaptive lexicon is necessary.

More than 100 hashtags were used to collect data over different time periods to insure a representative sample of crisis related images and captions. Figure 5 summarizes some of the hashtags we used to collect social media posts. Furthermore, we augmented our text data with informative tweets from the CrisisLexT26 dataset (Olteanu et al., 2015) and human and infrastructure damage tweets from the CrisisNLP dataset (Imran et al., 2016). We also augmented our image data with images from google search using the same keywords and hashtags.

Data filtering

After collecting the raw data which consists of text, images, and captioned images, we divide the data into one of five different damage categories, with an additional background non-damage class:

(1) Infrastructural damage: damaged buildings, wrecked cars, and destroyed bridges

(2) Damage to natural landscape: landslides, avalanches, and falling trees

(3) Fires: wildfires and building fires

(4) Floods: city, urban and rural

(5) Human: injuries and deaths

The end goal of the classification is to direct the appropriate resources for each situation. For example, identifying human casualties would prompt first responders to deploy ambulances and first aid kits, while identifying a fire would result in the mobilization of fire trucks. We based our categories on this objective: we want the classes to be descriptive of the damage rather than specify the source of the damage to aid in resource allocation. Ideally, it is preferable to have categories for each type of disaster that could arise, however this high specificity would sacrifice accuracy with current methods. These factors determined our choices for the categories.

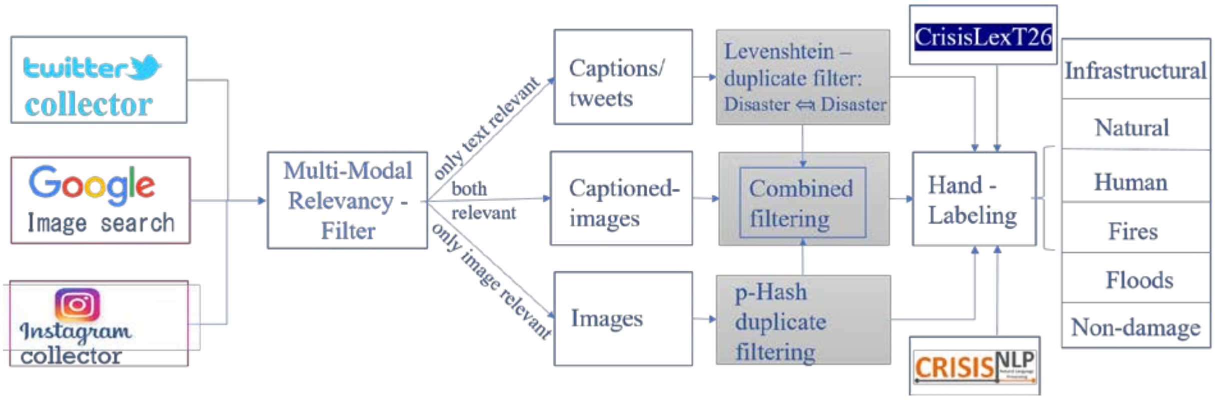


Figure 6. Data Collection and Processing Pipeline

Table 1. Dataset Statistics

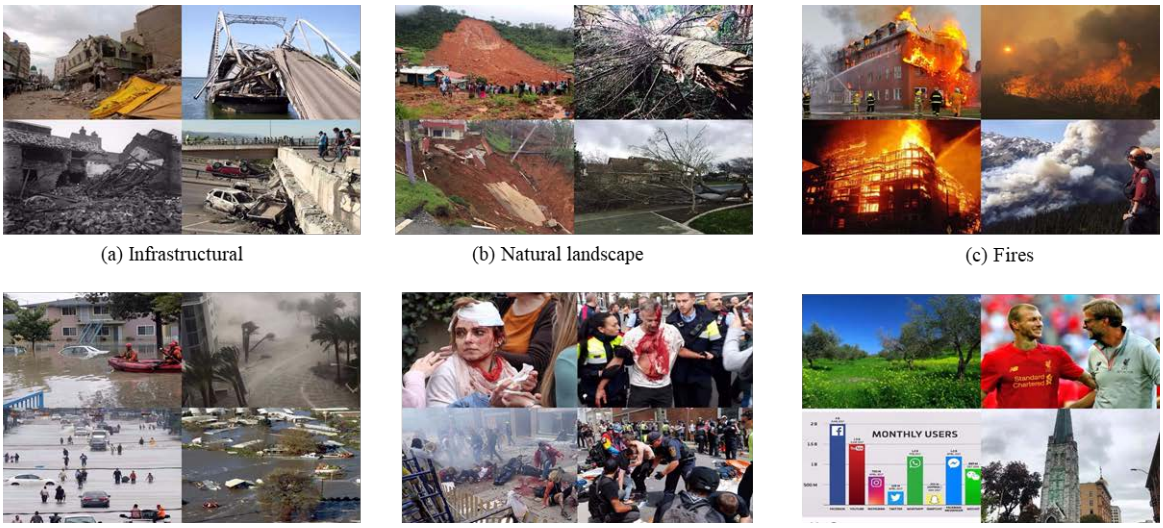
Label	Captioned Images	Text	Images without captions	Total
Fires	349	1819	856	3024
Floods	385	3685	1236	5306
Natural landscape	515	1876	882	3273
Infrastructural	1418	3547	3002	7967
Human	240	2910	1008	4158
Non-damage	2972	5194	3891	12057
Total	5879	19031	10875	35785

To speed-up the labelling process, we first filtered out irrelevant and non-informative posts, including advertisements, selfies or memes. To do this, we hand-labelled a subset of the collected posts as either relevant to disaster situations or not relevant. Then, we trained relevancy classifiers on these posts to recognize relevant vs. non-relevant posts. We applied this classifier to filter out the remaining data that are non-relevant. The remaining posts were manually labeled by a group of 5 annotators, a majority vote determined the final annotations. The inter-annotator disagreement rate was equal to 2.6%. The collected data belong to three mutually exclusive bins: text, images, and captioned images.

First, to filter out irrelevant text data, we employed a similar system to Tweet4act (Chowdhury et al., 2013). We trained a binary version of our text model to classify damage vs. non-damage, and then we deleted duplicate tweets/text by first computing a pairwise Damerau-Levenshtein distance between every pair of captions. The Damerau-Levenshtein distance represents the number of operations (insertion, deletion or substitutions) required to transform one string of text to another. If the distance is less than 10% of the respective lengths of the captions, we arbitrarily removed the shorter length text. Most duplicate captions occurred in the cases of retweets or automated bot tweeting.

Second, to filter out irrelevant images, we used the Image4Act system (Alam et al., 2017). We trained a binary classifier version of our image model again to classify damage vs. non-damage, and then we removed duplicate images by computing a perceptual hash (pHash) for the images. The pHash is a hashing algorithm designed for images; the pHash value for similar images will be almost equal and is resilient to multiple modifications of original images. Thus, we iterate over pairs of images and arbitrarily remove one of them if their pHash values are almost equal.

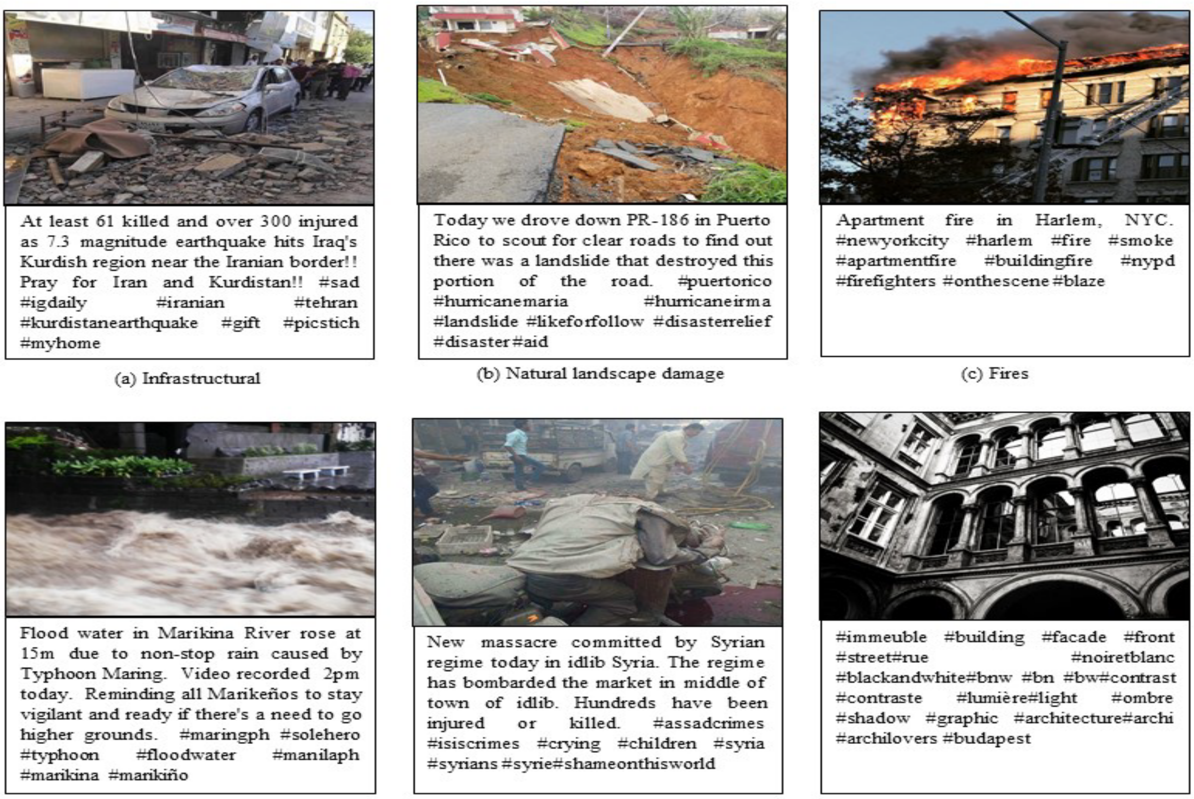
Finally, to filter out irrelevant captioned images, we used a multimodal decision based system. The caption is fed into the text relevancy model and the image into the image relevancy model. If both models agree on relevancy, then we move the captioned image into its corresponding category. Otherwise, the text and image are placed in separate categories. Identifying duplicate captioned images is similar: both signals need to be duplicates of the corresponding signals of another post to delete the captioned image. If only the image part of the post is a duplicate, we remove it and place the text into its bin, and vice-versa, to avoid situations where one sample is in the training set and another in the test set. This filtering process is depicted in Figure 6.



(a) Infrastructural (b) Natural landscape (c) Fires

(d) Floods (e) Human (f) Non-damage

Figure 7. Sample Dataset Images



At least 61 killed and over 300 injured as 7.3 magnitude earthquake hits Iraq's Kurdish region near the Iranian border!! Pray for Iran and Kurdistan!! #sad #igdaily #iranian #tehran #kurdistanearthquake #gift #picstich #myhome

(a) Infrastructural

Today we drove down PR-186 in Puerto Rico to scout for clear roads to find out there was a landslide that destroyed this portion of the road. #puertorico #hurricanemaria #hurricaneirma #landslide #likeforfollow #disasterrelief #disaster#aid

(b) Natural landscape damage

Apartment fire in Harlem, NYC. #newyorkcity #harlem #fire #smoke #apartmentfire #buildingfire #nypd #firefighters #onthescene#blaze

(c) Fires

Flood water in Marikina River rose at 15m due to non-stop rain caused by Typhoon Maring. Video recorded 2pm today. Reminding all Marikeños to stay vigilant and ready if there's a need to go higher grounds. #maringph #solehero #typhoon #floodwater #manilaph #marikina #marikiño

New massacre committed by Syrian regime today in idlib Syria. The regime has bombarded the market in middle of town of idlib. Hundreds have been injured or killed. #assadcrimes #isiscrimes #crying #children #syria #syrians#syrie#shameonthisworld

#immeuble #building #facade #front #street#rue #noiretblanc #blackandwhite#bnw #bn #bw#contrast #contraste #lumière#light #ombre #shadow #graphic #architecture#archi #archilovers#budapest

(d) Floods (e) Human (f) Non-damage

Figure 8. Sample Captioned Images

Dataset Characteristics

Based on the database collection procedure illustrated in the previous subsections, a home-grown database of 35,566 samples was collected from various sources. Table 1 summarizes the number of samples per class and the number of unimodal (text and images without captions) vs. bimodal samples (captioned images). A few sample images from each class are reported in Figure 7 and Figure 8.

Table 2. Image Classification Performance

Pretrained model	Validation Accuracy (%)	Test Accuracy (%)
Inception-v3	**84.86±1.27**	83.78±1.48
Inception-v4	84.41±2.00	**83.98±1.72**
VGG16	81.32±2.42	82.19±2.46
InceptionResnet-v2	84.16±1.58	83.61±0.82

Table 3. Text Classification Performance

Word Embedding	Number of filters	Validation Accuracy	Test Accuracy (%)
Word2Vec	256	**91.18±1.34**	**89.88±0.70**
Word2Vec	512	90.90±1.79	89.33±1.01
GloVe	256	89.56±1.60	89.22±0.69
GloVe	512	89.63±1.32	89.04±1.11

EXPIREMENTS

In this section, we evaluate unimodal deep learning classifiers and machine learning algorithms for the multimodal damage identification task. Experiments were run on a machine equipped with an Intel Core i7-6700 CPU with 4 cores and an Nvidia GTX1060 GPU with 6GB RAM running a Windows 8 operating system. The algorithms were written in Python based on the Tensorflow 1.3 package. The code and dataset are publicly available on Github[1]. The captioned images dataset was split into 4-folds to perform cross validation. Furthermore, each fold is split into a test set (70% of fold or 18% of original data) and a validation set (30% of fold or 7% of original data). The unimodal image and text models were trained on their respective unimodal datasets augmented with the training set of captioned images set for each fold. All results are averaged over the four folds, with the mean and standard deviation being reported.

Unimodal Classification

For the image model, we experimented with four different CNN architectures. All the models were pretrained on ImageNet and the weights of the networks' last layers were fine-tuned on our dataset. The networks were trained until the loss converged with the default hyperparameters. Table 2 reports the accuracy (mean±standard deviation) on the validation set for the various CNN models which were used to select the best model for the multimodal fusion. Additionally, we reported the test set accuracies for comparison with multimodal models. While all models achieved similar performance, Inception-v3 marginally outperformed other models on the validation set, and it was also the fastest model for training and prediction.

For the text model, we resorted to Kim's CNN architecture (Kim, 2014). We compared two different pretrained word embeddings: GloVe trained on 2 billion tweets and Word2Vec trained on a 3 billion words Google News corpus. For the network parameters, we used filters of size 3,4 and 5 with 256 or 512 filters for each size. Table 3 summarizes the performance of the CNN trained on the home-grown database portion reserved for training and tested on the validation set. The number of filters did not have a significant effect on the accuracies; default parameters of the network were effective for this text model. Word2Vec outperformed GloVe on the validation set by more than 1%, but only marginally outperformed it on the test set.

Based on the validation set results, we chose the Inception-v3 image model and the Word2Vec-256 text model as the basis for the multimodal classifiers discussed in the following sections.

Multimodal Classification

We first evaluated the DF multimodal approach. For the rule-based classifiers, we obtained validation set predictions of the best unimodal classifiers (Inception-v3 and Word2Vec-256) to compute the a priori weights for the weighted maximum decision rule model. Then, we applied the test set to the end-to-end system. For the simple maximum rule DF approach, we simply applied the test set to the unimodal classifiers and obtained the final class using the maximum decision rule. We trained the stacking classifiers on the training set, used the validation set for hyperparameter selection, and evaluated on the test set.

[1] https://github.com/husseinmz/multimodal-deep-learning-for-diaster-response

Table 4. Performance of DF Classifiers and Rules

DF algorithm	Test Accuracy (%)
Maximum decision rule	91.85±1.01
Weighted maximum decision rule	91.67±0.83
DFMC with ANN	92.30±1.09
DFMC with KNN	92.57±1.36
DFMC with SVM (gaussian kernel)	**92.62±0.89**

Table 5. Performance of FF Meta Classifiers

Classifier	Test Accuracy (%)
ANN	**92.60±0.77**
KNN	89.40±0.77
SVM (linear kernel)	92.27±0.53
SVM (gaussian kernel)	90.74±0.51

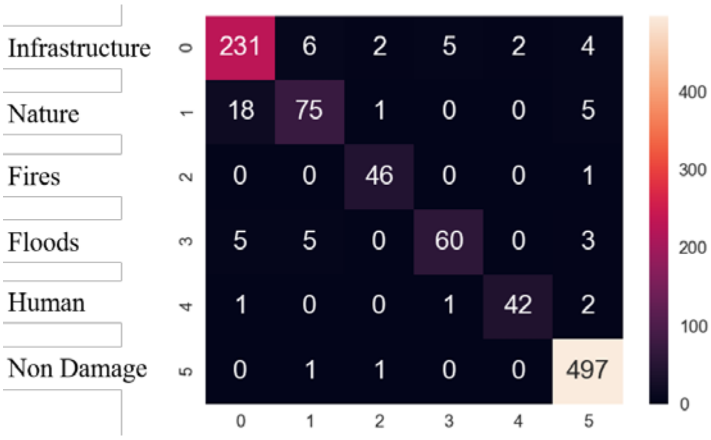


Figure 9. Confusion Matrix for Best Model

Table 4 reports the accuracies (mean±standard deviation) of various DF classifiers on the test set. All multimodal classifiers outperformed their unimodal counterparts by a minimum margin of 2% which hints at the effectiveness of using both modalities. Specifically, the DFMC with SVM outscored the best unimodal text model by 2.74%. We note that the stacking meta classifiers performed better than the decision rules. Furthermore, the weighted maximum decision rule scored slightly worse than the regular maximum rule which might be due to noisy class-estimates on the small validation set.

Next, we focused on the FF approach where the bottleneck feature i.e. the last layer generated from Word2Vec-256 and Inception-v3 networks were concatenated and used to train the meta classifier on the annotated images dataset. Multiple classifiers were compared including a 4-hidden layer ANN with 128, 64, 32, and 16 neurons per layer, SVM with linear and gaussian kernels, and KNN. Table 5 reports the accuracies on the test set, where the ANN classifier achieved the best accuracy of 92.60% and SVM with linear kernel was a close second. However, the best FF classifier and the best DF classifier achieved comparable accuracies (0.02% difference).

Figure 9 displays the confusion matrix of the best end-to-end model (DFMC with gaussian SVM whose input is obtained from the Inception-v3 image model and Word2Vec-256 text model). Our model could distinguish non-damage from damage data well; it achieved a 99.6% true positive rate for the non-damage class. Focusing damage related classes, the true positive rate per class ranged from 75.8% (for nature) to 97.9% (for fires). We notice that many of the nature images were classified as infrastructure which could be due to the existence of man-made structures such as houses or cars in the natural landscapes. The model correctly identified human casualties with a true positive rate of 91.3% but may have been affected by background scenes in images which often included buildings when misclassifying some of the data into infrastructure or non-damage.

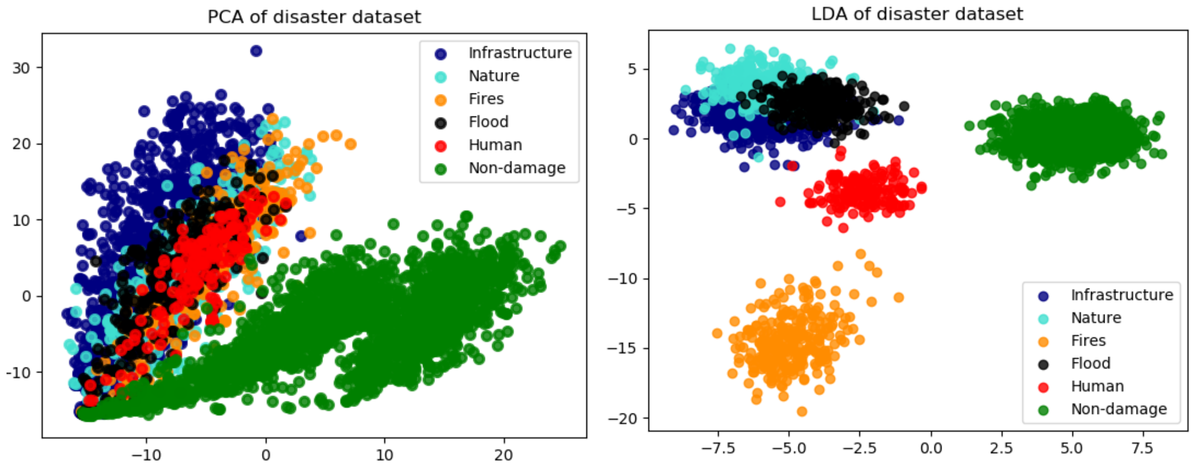


Figure 10. Results of PCA (Left) and LDA (Right) feature reduction

Class Separability Analysis

A principle component analysis (PCA) and latent discriminant analysis (LDA) were performed on our database to examine the separability of the investigated classes. The features obtained from the best image classification (Inception-v3) and text classification (Word2Vec-256) models were used as input to the PCA and LDA algorithms. Figure 10 plots the distribution of the data after performing PCA and LDA to reduce the dimensionality of the data to 2. The LDA plot shows that most of the classes are easily separated which confirms that our categories are distinct with only noticeable overlap existing between the flood and infrastructure damage data. The PCA plot shows that damage and non-damage classes are easily separable, which may explain why simple classifiers such as KNN performed well. This shows that the deep learning models learned class specific features well.

CONCLUSION

Identifying damage and human casualties in real time from social media posts is critical to providing prompt and suitable resources and medical attention, to save as many lives as possible. With millions of social media users continuously posting content, an opportunity is present for machine learning algorithms to learn a damage and human casualty recognition model. In this work, we present a multimodal deep learning framework that classifies social media posts into five damage related categories including infra-structure damage, natural damage and human casualties. A CNN model based on the Inception pretrained model is used to process images, and a CNN model fed with the word embedding model’s output is used to process text. Combining both inputs in a multimodal learning framework achieved an accuracy of 92.62%. Future work will investigate developing other multimodal deep learning models for a finer categorization and new online techniques for data collection and filtering.

ACKNOWLEDGEMENTS

This work was supported by the National Council of Scientific Research in Lebanon (CNRS-L). The authors would like to thank the reviewers for their insightful suggestions, and Elie Yakoub, Issa Issa, and Morgane El Ayle for their help in building our corpus.

REFERENCES

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10), 1533-1545.

Abel, F., Hauff, C., Houben, G. J., Stronkman, R., & Tao, K. (2012, June). Semantics+ filtering+ search= twitcident. exploring information in social web streams. In *Proceedings of the 23rd ACM conference on Hypertext and social media* (pp. 285-294).

Alam, F., Imran, M., & Ofli, F. (2017). Image4Act: Online Social Media Image Processing for Disaster Response. *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 601-604).

Alqhtani, S. M., Luo, S., & Regan, B. (2015). Fusing text and image for event detection in Twitter. *The*

International Journal of Multimedia & Its Applications, 7(1), 27.

Ashktorab, Z., Brown, C., Nandi, M., & Culotta, A. (2014, May). Tweedr: Mining twitter to inform disaster response. In *Proceedings of the international conference on information systems for crisis response and management*.

Baer, D. (2012). As Sandy Became #Sandy, Emergency Services Got Social. https://www.fastcompany.com/3002837/sandy-became-sandy-emergency-services-got-social

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798-1828.

Breen, W. A., & Ida, A. M. (2016). Implementation of Speedy Emergency Alert using Tweet Analysis. *Indian Journal of Science and Technology,* 9(11).

Caragea, C., McNeese, N., Jaiswal, A., Traylor, G., Kim, H. W., Mitra, P., ... & Yen, J. (2011, May). Classifying text messages for the Haiti earthquake. In *Proceedings of the 8th international conference on information systems for crisis response and management.*

Chowdhury, S. R., Imran, M., Asghar, M. R., Amer-Yahia, S., & Castillo, C. (2013, May). Tweet4act: Using incident-specific profiles for classifying crisis-related messages. In *Proceedings of the international conference on information systems for crisis response and management*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug), 2493-2537.

Conneau, A., Schwenk, H., Barrault, L., & LeCun, Y. (2016). Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*.

Cover, T. M., & Thomas, J. A. (2006). Elements of information theory 2nd edition.

Cresci, S., Cimino, A., Dell’Orletta, F., & Tesconi, M. (2015a, November). Crisis mapping during natural disasters via text analysis of social media messages. In *International Conference on Web Information Systems Engineering* (pp. 250-258). Springer, Cham.

Cresci, S., Tesconi, M., Cimino, A., & Dell'Orletta, F. (2015b, May). A linguistically-driven approach to cross-event damage assessment of natural disasters from social media messages. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 1195-1200).

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248-255).

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014, January). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (pp. 647-655).

Dos Santos, C. N., & Gatti, M. (2014, August). Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *COLING* (pp. 69-78).

Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning* (pp. 513-520).

Guillaumin, M., Verbeek, J., & Schmid, C. (2010, June). Multimodal semi-supervised learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 902-909).

He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In *European Conference on Computer Vision* (pp. 630-645). Springer, Cham.

Huh, M., Agrawal, P., & Efros, A. A. (2016). What makes ImageNet good for transfer learning? *arXiv preprint arXiv:1608.08614.*

Imran, M., Mitra, P., & Castillo, C. (2016). Twitter as a lifeline: Human-annotated twitter corpora for NLP of crisis-related messages. *arXiv preprint arXiv:1605.05894*.

Imran, M., Castillo, C., Lucas, J., Meier, P., & Rogstadius, J. (2014, May). Coordinating human and machine intelligence to classify microblog communications in crises. In *Proceedings of the international conference on information systems for crisis response and management.*

Iyyer, M., Boyd-Graber, J. L., Claudino, L. M. B., Socher, R., & Daumé III, H. (2014, October). A Neural Network for Factoid Question Answering over Paragraphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 633-644).

Jomaa, H. S., Rizk, Y., & Awad, M. (2016, November). Semantic and Visual Cues for Humanitarian Computing of Natural Disaster Damage Images. *Proceedings of the 12th International Conference on Signal-Image Technology & Internet-Based Systems* (pp. 404-411).

Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1746–1751).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, January). Recurrent Convolutional Neural Networks for Text Classification. In *AAAI*, 333 (pp. 2267-2273).

Le, H. T., Cerisara, C., & Denis, A. (2017). Do Convolutional Networks need to be Deep for Text Classification? *arXiv preprint arXiv:1707.04108*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning* (pp. 689-696).

Nguyen, D. T., Joty, S., Imran, M., Sajjad, H., and Mitra, P. (2016). "Applications of Online Deep Learning for Crisis Response Using Social Media Information". *In: arXiv preprint arXiv:1610.01030*.

Nguyen, D. T., Ofli, F., Imran, M., & Mitra, P. (2017). Damage Assessment from Social Media Imagery Data During Disasters. *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 569-576).

Olteanu, A., Castillo, C., Diaz, F., & Vieweg, S. (2014, June). CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. In *ICWSM*.

Olteanu, A., Vieweg, S., & Castillo, C. (2015, February). What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (pp. 994-1009).

Olteanu, A., Castillo, C., Diaz, F., & Kiciman, E. (2016). Social data: Biases, methodological pitfalls, and ethical boundaries.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532-1543).

Poria, S., Cambria, E., Howard, N., Huang, G. B., & Hussain, A. (2016). Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing*, 174, 50-59.

Potts, L., Seitzinger, J., Jones, D., & Harrison, A. (2011, October). Tweeting disaster: hashtag constructions and collisions. In *Proceedings of the 29th ACM international conference on Design of communication* (pp. 235-240).

Rhodan, M. (2017). 'Please Send Help.' Hurricane Harvey Victims Turn to Twitter and Facebook. http://time.com/4921961/hurricane-harvey-twitter-facebook-social-media/

Sakaki, T., Okazaki, M., & Matsuo, Y. (2013). Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4), 919-931.

Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 806-813).

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014, April). Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web* (pp. 373-374).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011, July). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 151-161).

Srivastava, N., & Salakhutdinov, R. R. (2012). Multimodal learning with deep boltzmann machines. In Advances in neural information processing systems (pp. 2222-2230).

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1), 1929-1958.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *AAAI* (pp. 4278-4284).

Techradar (2013). Dealing with disaster: how social media is helping save the world. https://www.techradar.com/news/internet/dealing-with-disaster-how-social-media-is-helping-save-the-world-1203809/

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156-3164).

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2), 241-259.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning* (pp. 2048-2057).

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923.*

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (pp. 649-657).

Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Zhou, Z.-H. (2009). Ensemble learning. In *S. Z. Li, editor, Encyclopedia of Biometrics*, pp. 270–273. Springer, Berlin.

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2017). Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*.