A Computationally Efficient Multi-modal Classification Approach of Disaster-related Twitter Images

Yara Rizk
Electrical and Computer Engineering Department,
American University of Beirut
Beirut, Lebanon
yar01@mail.aub.edu

Hadi Samer Jomaa
Electrical and Computer Engineering Department,
American University of Beirut
Beirut, Lebanon
hsj04@mail.aub.edu

Mariette Awad
Electrical and Computer Engineering Department,
American University of Beirut
Beirut, Lebanon
mariette.awad@aub.edu.lb

Carlos Castillo
Universitat Pompeu Fabra
Spain
chato@acm.org

ABSTRACT

When natural disasters strike, annotated images and texts flood the Internet, and rescue teams become overwhelmed to prioritize often scarce resources, while relying heavily on human input. In this paper, a novel multi-modal approach is proposed to automate crisis data analysis using machine learning. Our multi-modal two-stage framework relies on computationally inexpensive visual and semantic features to analyze Twitter data. Level I classification consists of training classifiers separately on semantic descriptors and combinations of visual features. These classifiers' decisions are aggregated to form a new feature vector to train the second set of classifiers in Level II classification. A home-grown dataset is gathered from Twitter to train the classifiers. Low-level visual features achieved an accuracy of 91.10% which increased to 92.43% when semantic attributes were incorporated. Applying such data science techniques on social media seems to motivate an updated folk statement "an ANNOTATED image is worth a thousand words".

CCS CONCEPTS

• **Computing methodologies → Information extraction**; **Scene understanding**; **Feature selection**; *Supervised learning by classification*;

KEYWORDS

Humanitarian computing, damage, infrastructure, nature, bag of words, low-level visual features, multi-modal classification

ACM Reference Format:
Yara Rizk, Hadi Samer Jomaa, Mariette Awad, and Carlos Castillo. 2019. A Computationally Efficient Multi-modal Classification Approach of Disaster-related Twitter Images. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3297280.3297481

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297481

1 INTRODUCTION

Natural disasters occur frequently, on an average of 388 disasters annually, causing economic damages worth an average of 156.7 billion US dollars [9]. A main obstacle to mitigating the extent of these damages is the lack of advanced warning [10]. In the wake of a crisis, response teams are flooded with help requests and may misdirect resources due to inaccurate information.

With the worldwide spread of social media networks and over 3.174 billion subscribers in 2015 [27], written and visual information can be communicated as fast as a click of a button. Utilizing this information could relieve dedicated man power from tedious identification required before initiating a response, and could lead to faster response, better resource management and prioritization.

Humanitarian computing spans a large field of applications including spreading awareness and alerts about possible natural disasters and information-processing methods to extract actionable information from social media such as the Artificial Intelligence for Disaster Response (AIDR) [14]. While some of the crisis-related topics suggested by Imran et al. [11] may or may not typically include images that can be properly exploited, other topics are more commonly associated with images from which we can assess the existing type of natural disaster damage. Two types of damage are considered: *built-infrastructure damage* and *nature damage*. The former is defined as losses to the built environment (e.g. buildings, bridges, roads); the latter includes losses to the natural environment due to hazardous events (e.g. trees, forests, farm land).

Thus, a natural research question emerges: how can image processing and natural language processing (NLP) be leveraged - separately or combined - in a data science framework to provide a smart real-time disaster data classifier? Can computationally inexpensive image processing approaches with some semantic analysis be helpful to first-aid responders in assessing the damage from a given natural disaster? To investigate these research questions, we extend the work presented in [15] by compiling a larger dataset and proposing a novel multi-modal approach that merges generic semantic attributes extracted from disaster-related Twitter messages with low-level visual features extracted from the corresponding

Figure 1: Illustration of the application

images to aid in humanitarian computing. This algorithm would be integrated into the humanitarian computing workflow in figure 1.

A two-level multi-modal classification scheme is proposed. The first level, referred to as Level I classification hereafter, trains classifiers on visual features and semantic features separately. Computationally inexpensive low-level visual features include color, shape and texture features. Semantic features are formed based on bags of words (BoW) created using WordNet synsets [19]. While deep learning approaches may outperform feature engineering approaches, their computational cost from an energy consumption perspective may not be suitable for our application. Running on mobile phones with limited access to power sources, the proposed algorithm should be energy efficient to benefit users. In Level II classification, the scores of the outcome from Level I classification are aggregated to form a new multi-modal feature vector, which is used to train a new classifier. In this sense, Level I classifiers act as a kernel that results in new features to train classifiers during Level II classification. A corpus from Twitter is collected to validate our approach. We focus on Twitter content as a social media outlet due to a few factors. Twitter is one of the most widely used social media networks, with approximately 310 million monthly active users, 83% of whom are on mobile devices.

The main contributions of this work include (1) a multi-modal two-stage damage classification framework which achieved 92.43% accuracy; (2) a computationally efficient visual feature vector to represent tweeted images which outperformed state-of-the-art results in the literature on outdoor city vs. natural landscapes; (3) a dense semantic descriptor to represent tweets; and (4) a home-grown damage database containing text and images from tweets.

Next, section 2 presents related work in the field of scene understanding and NLP for humanitarian computing. The proposed methodology is detailed in section 3. Section 4 presents the dataset creation workflow, whereas section 5 presents the experimental results, before concluding with final remarks in section 6.

2 LITERATURE REVIEW

Humanitarian computing frameworks that mine social media have been developed to aid first responders, governments and decision makers in times of crisis [12]. Some of the existing humanitarian computing NLP frameworks and multi-modal classification approaches are surveyed next.

2.1 Humanitarian Computing Applications

Humanitarian computing applications range from event detection [1] to disaster mapping [17] and generating alerts [5]. In this work, we take a subfield of humanitarian computing known as actionable information extraction where social media posts are automatically processed to extract useful information for first responders. More specifically, we focus on damage identification.

End-to-end systems such as AIDR [13], EMERSE [6] and Tweedr [4] monitor social media sites, collect and classify posts to extract actionable information for first responders by identifying damage related topics, mainly in text. Image4Act, an end-to-end image processing tool, assessed the severity of infrastructure damage in images posted on social media using deep learning [2]. Similarly, [22] assessed the degree of damage in images using convolutional neural networks (CNN). Mouzannar et al. [20] proposed a deep learning multi-modal classification of disaster-related social media posts. CNN were used to classify process raw images and text before classifying social media posts into one of six classes using softmax layers. Jomaa et al. [15] also adopted a multi-modal approach with feature engineering instead of deep learning to reduce the computational complexity of the algorithm, making it suitable for edge computing scenarios when the cloud is not accessible.

2.2 Feature Fusion

Multiple references have developed multi-modal models to improve classification performance. Deschacht et al. [8] incorporated WordNet-formed synsets of salient words from image annotations to learn a probabilistic visual object recognition model. Alqhtani et al. [3] aggregated BoW semantic descriptors, with a set of texture and color visual features to detect events from Twitter posts. Poria et al. [23] fused audio features with both visual and textual information for multi-modal sentiment analysis. Multi-modal deep learning approaches include Ngiam et al. [21] who trained deep neural networks using audio and visual data to learn a shared representation, and Srivastava et al. [26] who learned a joint image-text representation by training a deep Boltzmann machine.

3 MULTI-MODAL TWO-STAGE CLASSIFICATION FRAMEWORK

3.1 Overall Workflow

In this work, we propose a multi-modal two-stage classification workflow, presented in figure 2. Once a tweet, containing textual and visual aspects, is retrieved, the text and image are extracted and processed independently. Computationally inexpensive, low-level visual features including shape, color, texture and energy are extracted to train a classifier. In parallel, binary semantic descriptors are derived by projecting the text onto our built BoW to train another classifier. The combined outputs of these Level I classifiers are

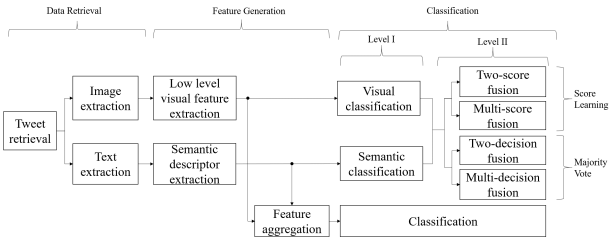

Figure 2: Multi-modal two-stage classification workflow

then used to train Level II classifiers. In this second classification phase, two approaches are compared: score learning and majority vote. The former uses the probability of belonging to each class as a feature vector while the latter builds the feature vector from the binary decisions. Finally, a classifier, referred to as feature aggregation in what follows, is also trained using the multi-modal feature vector (concatenation of visual and semantic feature vectors).

3.2 Visual Feature Extraction

Representing an image via proper low-level features is one of the most challenging steps in scene classification. Table 1 summarizes our features that encompass distinct aspects of nature and built-infrastructure scenery [15], specifically color, shape, texture, and energy. These features, extracted from 256×256 images, are computationally inexpensive, dense, and extensively used in the field of scene understanding.

(1) **Red-green-blue (RGB) histogram** is based on the red, green and blue color channels. We adopt a 256-bin size per channel to capture the maximum number of pixel variations.
(2) **Hue-Saturation-Intensity or Value (HSI/V) histogram** is an 84-bin histogram (36 for hue, 32 for saturation and 16 for intensity) that is scale and shift-invariant to light intensity. It provides information about the color content in images.
(3) **Gradient direction histogram** quantifies the distribution of gradient direction of pixels. This feature discriminates our two classes since urban scenery contains more edges than natural images. We set the bin sizes to 720, which corresponds to gradients of 0.5 degrees of resolution, to capture the fine gradients in the image. In figure 3(b), the image shows scaled values of the gradient direction at every pixel. At the boundaries of the building and the rubble one can notice that the gradient direction value is of similar color.
(4) **Gray-Level Co-Occurrence Matrix (GLCM)** is a statistical method that captures the texture of an image using spatial relationships between pairs of gray-value intensity pixels, as shown in figure 3(c). Correlation, contrast, homogeneity, energy and entropy, for specified displacements or offsets in the image, are features derived from GLCMs.
(5) **Gabor** captures the energy of an image using fast Fourier transform to the response at various scales and orientations. This helps capture object patterns and edges at varying frequencies and orientations, as in figure 3(d).
(6) **GIST** summarizes the gradient information (scales and orientations) in an image, which provides a rough description (the gist) of the scene (figure 3(e)). It is based on a set of perpetual dimensions, mainly naturalness, openness, roughness, expansion and ruggedness of the image.

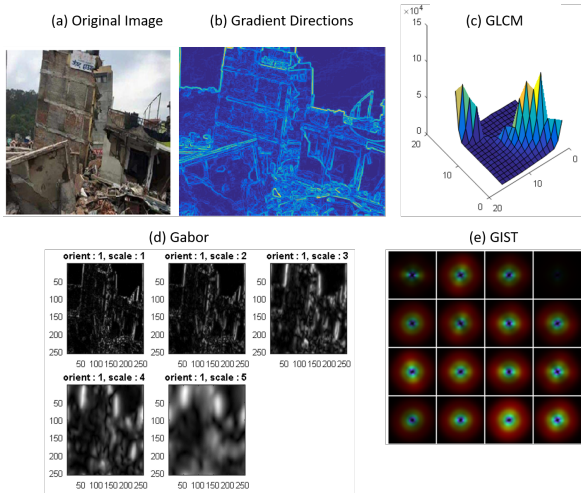

Figure 3: An example of the visual features

Table 1: Visual Feature Characteristics

Feature Type	Feature	Vector Size
Color	RGB Histogram	768
	HSV Histogram	84
Shape	Gradient Directions Histogram	720
Texture	GLCM	168
Energy	Gabor	30
	GIST	512

3.3 Semantic Feature Extraction

To improve the classification of images, a semantic understanding of the annotations is proposed. Common approaches to build semantic descriptors include using word embedding and deep learning which is computationally expensive or using binary vectors that determine the existence of BoW and N-grams (sequence of words) in a sentence but produce sparse vectors. In this work, we propose a workflow to generate a dense semantic descriptor based on the creation of domain specific BoW as follows.

3.3.1 BoW Creation. A list of distinct words, we call *Terms*, is compiled from the collected data, to determine the most common words used to describe damage in images. Table 2 provides examples of the most frequent *Terms* we found. WordNet [24], a large English lexical database, generates synonyms of these *Terms*. It groups words into sets of cognitive synonyms (Synsets), or semantic levels, based on conceptual-semantic and lexical relations including synonymy, meronymy, and antonymy. Hence, the number of Synsets to which a word belongs to differs from one to another; we considered 50

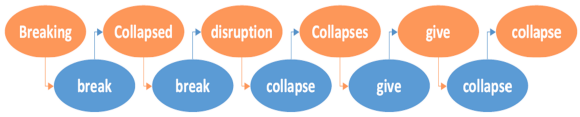

Figure 4: Combined (orange) and Joining (blue) Words

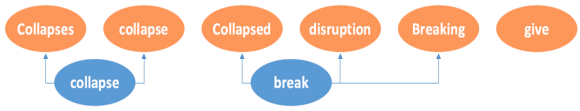

Figure 5: Filtered Combined Words

Synsets, since minimal change was observed between Synset 49 and 50. Table 2 shows examples of retrieved Synsets for sample *Terms*. Hereafter, Data01 refers to the semantic descriptors obtained by projecting the messages on the BoW based on the 1st Synset, Data05 from the 5th Synset and so forth.

Next, related *Terms* are grouped in BoW by iteratively looking for mutual synonyms at every semantic level between pairs of *Terms*. An example is shown in figure 4 where orange ovals represent *Terms*, blue ovals represent mutual synonyms or joining words, and arrows indicate a synonymous or root word relationship. However, some words in the Boo might be unrelated, e.g. "give" and "breaking". To keep the list of combined *Terms* diverse, yet relevant, an additional step is performed. Every set of combined *Terms* sharing at least one synonym are grouped together and separated from the set previously established. Hence, the set of *Combined Words* in figure 4, is divided into two distinct sets. The first corresponds to "breaking" and its *Combined Words'* synonyms, presented in figure 5, whereas "give" is excluded from the *Combined Words* and its synonyms are removed from the separate set.

Table 2: Sample Base Words and Synonyms

Terms	Freq	1st Synset	5th Synset	10th Synset
earthquake	609	quake, temblor, seism	*empty*	*empty*
damage	387	harm, impairment	harm, hurt, scathe	*empty*
quake	370	earthquake, temblor, seism	tremor	*empty*
help	78	aid, assist, assistance	help, assist, aid	help
death	160	decease, expiry	*empty*	*empty*
hits	157	hit	hit	hit, strike
storm	70	violent storm	force	*empty*

3.3.2 Semantic Descriptor Creation. Now that the BoW have been formed, image annotations are projected onto the BoW to create the semantic descriptors based on the workflow shown in figure 6. Words are parsed, then their membership to the BoW is assessed. If a word belongs to one of our BoW, it is replaced by the BoW's

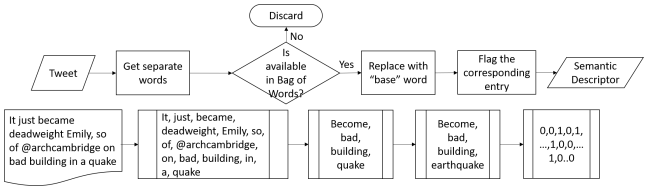

Figure 6: Semantic Descriptor Extraction Workflow

base word and its corresponding flag is set. This transforms the annotations into combinations of base words (filtered annotations), instead of different (related) words. As a result, we obtain a binary semantic descriptor, with a vector size equal to the number of the BoW, which is still sparse due to the shortness of tweets compared to the large number of BoW.

The semantic descriptor is made denser by eliminating BoW and their lead words from the feature vector if the probability of occurrence of the base words in both classes is less than 5% (chosen heuristically); examples are included in Table 3. Increasing this threshold increases the density (decreases the length) of the semantic descriptor but might cause the classifiers to miss subtle words that distinguish the classes, evident in the experimental results where lower accuracy was achieved for higher thresholds. To illustrate this idea, Table 4 summarizes the length of the semantic descriptors at different thresholds for the first 10 semantic levels.

This approach does not take into consideration negated terms, nor leverage prior knowledge regarding prominent terms. A comparison to descriptors generated by the Word2Vec word embedding method is performed in section 5.

Table 3: Probability of Terms per Class

Sample Term	*Built-infrastructure Damage*	*Nature Damage*
rise	0.1170	0.0517
death	0.1486	0.0369
photograph	0.0669	0.0627
kill	0.1216	0.0590
home	0.0557	0.0664

Table 4: Semantic Descriptor Length vs. Threshold

Semantic Level	5%	10%	15%
Data01	32	12	6
Data02	34	13	9
Data03	30	11	8
Data04	32	11	7
Data05	32	12	8
Data06	32	12	7
Data07	30	11	8
Data08	27	11	8
Data09	27	11	6
Data10	27	10	6

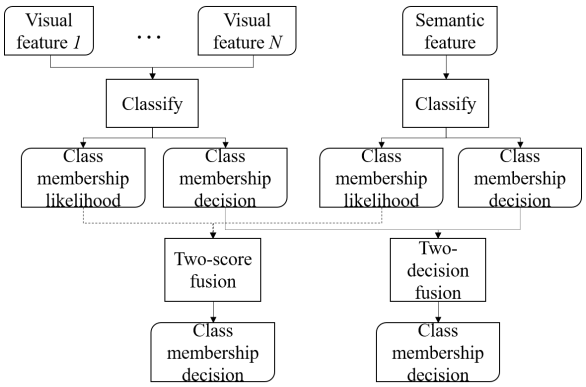

Figure 7: Two-Score Fusion and Two-Decision Fusion

3.4 Two-Stage Classification

Once features were extracted, a multi-modal two-stage classification approach is adopted to distinguish between *built-infrastructure damage* and *nature damage*. In Level I, the visual and semantic feature vectors independently train a set of classifiers to perform said classification on images. Since multiple visual feature types are extracted, we consider two methods of training classifiers based on these visual features. In the first approach, all the visual features are aggregated in one vector and used to train on classifier, as shown in figure 7. As a result, two classifiers are trained independently, one for semantic features and one for visual features. In what follows, the prefix *two-* is added to refer to this approach. In the second approach, each type of visual features trains a classifier, resulting in N independently trained classifiers from visual features, as shown in figure 8. Therefore, $N + 1$ classifiers are trained independently and allows us to study the contribution of type of feature. The prefix *multi-* is added to refer to this approach.

In Level II, two approaches can be adopted as well to train the second set of classifiers. The first approach, referred to as score learning or score fusion, uses the class membership likelihood (probability) produced by Level I classifiers to train a new classifier on the binary classification task. The feature vectors in Level II contain continuous values. The second approach, referred to as majority vote or decision fusion, combines the individual classifier's class membership decisions to produce a final decision by taking the class with the higher number of votes. In summary, four methods are proposed: two-score fusion, multi-score fusion, two-decision fusion, and multi-decision fusion. We compare them to the more conventional multi-modal classifiers trained on aggregates features, instead of classifier outputs.

4 DATABASE

4.1 Home-grown Database

The home-grown database was solely collected from the Twitter feed. First, keywords such as earthquake, damage, disaster, crisis, flood, etc., were queried to retrieve corresponding tweets (both text and images when available). These tweets were posted between February and May 2016, when earthquakes hit Nepal, Chile, and Japan, and floods hit Kenya. Manual filtering removed redundant (retweets) or irrelevant images (did not contain damage).

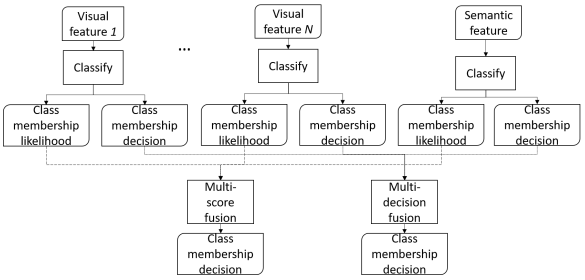

Figure 8: Multiple-Score and Multiple-Decision Fusion

Figure 9: Database Samples: (A) *Built-Infrastructure Damage*, (B) *Nature Damage*

Table 5: Database Statistics

	Class 1	Class 2	Database size	Imbalance Ratio
Home-grown	1077	271	1348	3.975
SUN	740	256	996	2.891

The collected data is manually labeled for supervised learning. A high inter-annotator agreement (for 3 annotators) was observed despite the broad and fuzzy nature of the term "damage" whose visual representation is difficult to quantify. For example, a fallen tree and a broken wall are considered damage. Three graduate students majoring in artificial intelligence independently labeled the data based on the visual content. The final labels were determined by majority vote. Figure 9 shows sample images and annotations; *Built-infrastructure damage* images (Class 1) were more dominant in the retrieved tweets than *nature damages* (Class 2). Table 5 summarizes the database statistics.

4.2 SUN Database

The SUN database [28] consists of images, gathered from several search engines. It encompasses 397 scenes labeled based on the object content. Since the proposed approach handles infrastructure and nature damage, the closest categories in the SUN database were *city* vs. *landscape*. The former consisted of "rubble, office building, city and building façade", whereas the latter consisted of "archaeological excavation, bog, forest and forest road". No preprocessing

was performed except for re-sizing images to 256×256. Database statistics are also summarized in Table 5.

5 EXPERIMENTAL RESULTS

5.1 Experimental Setup

All simulations were executed in MATLAB R2015b on an Intel(R) Core(TM) i7-4700MQ and 2.4GHz CPU with Windows 10 64-bit operating system. We adopted 5-fold cross-validation. F-measure ($\beta = 1$) and accuracy metrics were used to compare classifiers. Negative class (Class 2) represented *nature damage* whereas positive samples (Class 1) represented *built-infrastructure damage*. Features were normalized into a Gaussian distribution with zero-mean and standard deviation of 1. Ensemble learning was obtained by fitting 20 weak classifiers via RUSBoost [25]. ANN architecture consisted of one hidden layer with the number of neurons equal to that of the descriptor dimension. A grid search was performed to find the optimal classifier hyper-parameters.

5.2 Level I Classification

5.2.1 Visual feature classification. First, we consider the performance of classifiers trained using the low-level visual features. The six distinct features discussed in Table 1 generate $\sum_{k=1}^{6} \binom{6}{k} = 63$ possible feature vector combinations by choosing a subset of features to train and test the visual classifiers. Figure 10 and 11 report the accuracy and f-measure of the six combinations with the best performance on the home-grown and SUN databases, respectively. The statistics, are the highest for every value of pairing, i.e. Plus 1 for the Grad feature is the highest among all other pairs of the Grad feature with other features, and so forth. In the both data sets, the accuracy increases with the number of aggregated features, and RGB features exhibiting the highest increase. The best accuracy was obtained with the kernel SVM using Gabor-GIST-Gray-HSV (Plus 3) equal to 90.65% and a precision of 72.50% on the home-grown dataset. In general, RGB features' poor performance could be contributed to the fact that both classes could contain a wide spectrum of colors. For example, green may be more common in *nature damage* images than gray but *built-infrastructure damage* images may also contain green colors. On the SUN database, an accuracy of 96.25% and precision of 93.17% was achieved using a kernel SVM classifier which shows the effectiveness of our proposed features in distinguishing between city and nature landscapes. In the literature, an accuracy of 93.5% and 92.7% on city and nature landscapes, respectively, using local binary patterns with linear logistic regression in [7]. Table 6 summarizes the running time of each visual feature extraction algorithm. Computing the gradient directions is the most expensive, taking almost 1.2 seconds, whereas the color histograms are the fastest, taking less than 0.15 seconds.

When we experimented with a pre-trained Inception (version 3) deep learning model fine tuned on the training set, a test accuracy of 95.5% and f-measure of 93.9% was achieved. While this is higher than the best low-level feature workflow, the computational cost during testing is significantly higher. Specifically, Inception performs 5.72 billion operations for 299×299 images [29] and 23.8 million floating point numbers saved in memory. On the other hand, the low level feature extraction algorithm requires less than 5 million operations and requires 2,282 floating point numbers saved in memory per image. For classification, shallow learners do not require more than a few million operations at test time as well. For example, SVM require approximately $2282N_{sv}$ floating point numbers to save the model, where N_{sv} is the number of support vectors and was in the order of a few hundred vectors for our model. Less memory and computations reduce energy consumption which allows a device's memory to last longer when power sources are scarce. In the context of disaster management where decisions need to be pseudo real time, a computationally effective model similar to our flow seems to be a better compromise when compared to deep models.

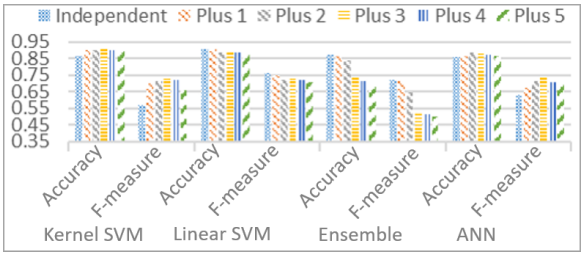

Figure 10: Performance of visual feature combinations on the homegrown dataset

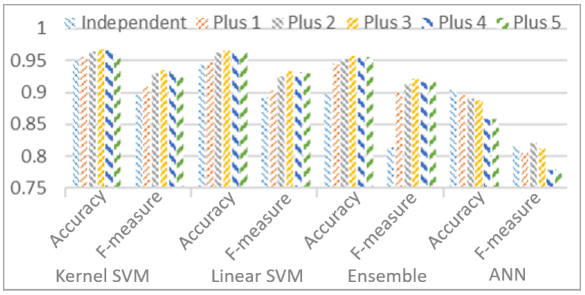

Figure 11: Performance of visual feature combinations on the SUN database

Table 6: Visual Features Extraction Running Time

Feature	Time (sec)
RGB Histogram	0.148
HSI Histogram	0.135
Gradient Directions Histogram	1.104
GLCM	0.448
Gabor	0.253
GIST	0.437

5.2.2 Semantic descriptor classification. Table 7 summarizes the f-measure of the semantic descriptors built from the first 10 semantic levels (Data01 till Data10). The best performance is achieved with Data07. With an accuracy of 83.53% for Data09, the f-measure and precision are far better, 39.87% vs 29.76% and 75.10% vs 56.96% respectively. The low sensitivity values indicate that several negative class entries are being misclassified as positive. Linear SVM

Table 7: F-measure of semantic descriptor classification

Features	Dimension	Kernel SVM	Linear SVM	Ensemble	ANN
Data01	32	0.37	0.35	0.52	0.23
Data02	34	0.36	0.35	**0.53**	0.00
Data03	30	0.33	0.32	0.50	0.00
Data04	32	0.37	0.32	0.48	0.27
Data05	32	0.38	0.32	0.48	0.17
Data06	32	0.36	0.32	0.49	0.40
Data07	30	0.40	0.32	0.52	0.00
Data08	27	0.30	0.24	0.48	0.35
Data09	27	0.29	0.31	0.48	0.25
Data10	27	0.29	0.31	0.50	0.32
Word2Vec	1	0.10	0.00	0.30	0.25
Word2Vec	2	0.00	0.00	0.38	0.00
Word2Vec	5	0.05	0.00	0.37	0.18
Word2Vec	10	0.10	0.35	0.38	0.25

accuracy is almost consistent with a standard deviation of 0.3%. Although ensemble learning resulted in the highest f-measure, its accuracy is incredibly low in comparison to the other classifiers. RUSBoost was specifically employed to accommodate the imbalance in the dataset which is 3.971. Entries from the "major" class are under-sampled, and then learning is boosted using AdaBoost.

To better assess the performance of our proposed semantic descriptor, we compare it to a semantic descriptor generated by Word2Vec [18], which utilizes continuous BoW and skip-gram architectures to compute vector representations of words. The raw semantic data is represented using numerical vectors of dimension 1, 2, 5 and 10 generated by a Word2Vec model pre-trained on the first billion characters from Wikipedia [16]. While higher dimensions (up to 300) are commonly used in the literature, we chose values that generated semantic vectors close in dimension to our proposed features. Table 7 f-measure of the Word2Vec features vectors. The accuracy improved as the dimensionality of the vector increased. Representing the words with one or two values, Word2Vec01 and Word2Vec02 respectively, prevents the classifiers from properly identifying the true negatives, which can be deduced from the lack of f-measure values among three out of four classifiers used. Finally, our proposed semantic descriptor outperforms Word2Vec's feature vectors for all classifiers.

5.3 Level II Classification

The four algorithms that were presented for Level II classifiers are compared to each other and feature aggregation method.

5.3.1 Decision Fusion. First, considering the decision fusion two-stage approach, figure 12 reports the difference between the majority vote of kernel SVM classifiers trained on Gabor, Gist, Gray, HSV and Data07 compared to the classifiers trained on visual features only and semantic features only. A negative value implies that the unimodal classifier outperformed its multi-modal counterpart. The

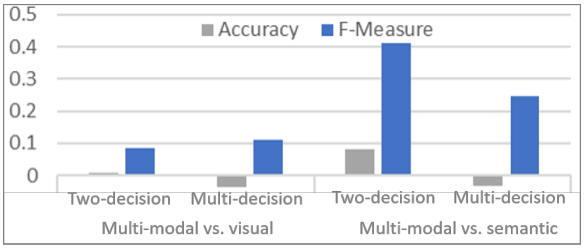

Figure 12: Comparing multi-modal decision-fusion classifiers to unimodal classifiers

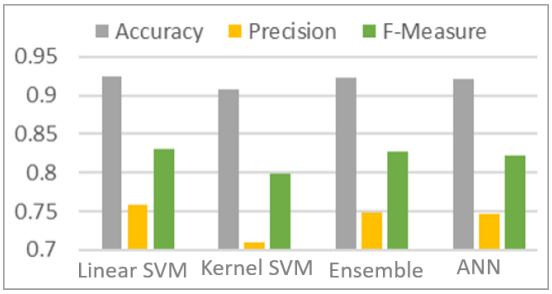

Figure 13: Performance of two-score fusion classifiers

two-stage classification approach, whether two-decision or multi-decision, improved the f-measure which highlights the importance of multi-modal features.

5.3.2 Score Fusion. Score fusion is tested on multiple classifiers in Level II trained on feature vectors formed from kernel SVM Level I outputs. Figure 13 reports the performance of four classifiers. Linear SVM achieved the highest accuracy, precision and f-measure. In a sense, the independently trained SVMs' act as a kernel, reducing the feature vector's dimension. A linear SVM in Level II may have been best because Level I classifiers transformed the data to a linearly separable space. Generally, the performance of the Level II classifier heavily relies on the performance of Level I classifiers. As above, a negative value implies that the unimodal classifier outperformed its multi-modal counterpart.

Figure 14 compares the difference between two-score and multi-score fusion (D1), and two-score fusion and feature aggregation (D2). Two-score fusion was better (positive difference) than feature aggregation across all classifiers. However, it was not always better than multi-score fusion, performing worse on kernel SVM (negative value). Comparing score fusion and feature aggregation to classifiers trained on corresponding visual features only, figure 15 clearly shows the improved performance of the multi-modal approach for all three feature fusion approaches; two-score fusion, multi-score fusion and feature aggregation each saw up to 13%, 15% and 19% improvement, respectively.

In summary, Table 8 reports the best set of features for every Level II classifier when a kernel SVM is used in Level I, for feature aggregation, two-score and multi-score fusion. Gray visual features improved the performance regardless of the classifier algorithm since they provide texture information which discriminates

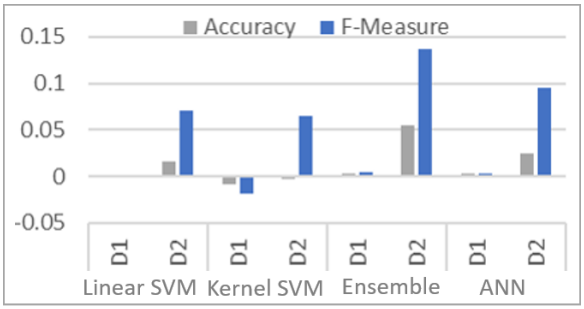

Figure 14: Comparing the accuracy of two-score fusion to multi-score fusion (D1) and feature aggregation (D2)

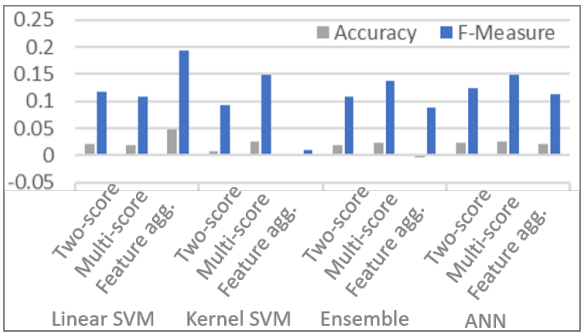

Figure 15: Difference in accuracy between different multi-modal classifiers and their corresponding visual feature-based unimodal classifiers

between the two classes. For semantic descriptors, Data01, Data03 and Data06 performed better than other semantic vectors. The best accuracy (92.43%) was achieved by a two-score fusion approach with a linear SVM. In general, two-score fusion outperformed feature aggregation regardless of the type of classifier adopted.

5.4 Error Analysis

Next, we take a closer look at the misclassified data based on the visual, semantic and combined features.

5.4.1 Level I Classification: Visual Features. The number of false positives (FP) was significantly larger than false negatives (FN) for Level I classifiers trained on visual features only (105 vs. 21). Figure 16 contains some examples of FP, images that were supposed to be classified *nature Damage*, but weren't. These FP contain one of four main characteristics that may have led to their misclassification: (A) images share an open field view with garbage (visually scattered colors); (B) images contain houses in an urban location with slight damage to nature components (broken trees); (C) images contain broken branches in nature scenery as opposed to the countryside; and (D) images contain groups of people in the damaged locations. In figure 17, FN samples could be categorized into two main clusters: (A) images include *infrastructure damage* to property in a nature environment (the abundance and continuity of the green color is

Figure 16: False Positives

Figure 17: False Negatives

evident); (B) images share a "flat" ground in nature environment (absence of chunks of rubble or broken buildings).

5.4.2 Level I Classification: Semantic Features. Table 9 reports some examples of misclassified sets of words by the classifiers trained on semantic features only. These sets of words were extracted from tweets that belong to both classes, since the true positive (TP) and true negative (TN) corresponding to these phrases are both non-zero. The ratio, taken to be TP over TN, reflects the relative frequency of a phrase belonging to Class 1 over Class 2. For example, the phrase "bring, damage, storm" is associated with five *nature damage* images and two *built-infrastructure damage* images, resulting in a ratio of 0.4. Due to its more frequent occurrence in Class 2, the semantic classifier is biased towards the negative class, and subsequently misclassified two instances of this phrase. These examples show that there is no fixed set of phrases for every class that can be used as a template for training. In some cases, phrases are present in both classes; this large overlap necessitates a second modality (images) to distinguish these classes.

5.4.3 Level II Classification. A two-score fusion two-stage classifier using linear SVM. Some misclassified images and their corresponding annotations are shown in figure 18 and 19. FN could be divided into two groups. Group 1 images contained rubble and broken houses and annotations contained common words like "damage, death, earthquake, rise". Group 2 images exhibited partially damaged buildings that were still standing upright; annotations contained "earthquake, damage, lay waste to". FP were also divided

Table 8: Best Classifier Performances Summary

		Visual Features	Semantic Descriptor	Accuracy (%)
Kernel SVM	Two-score fusion	Gist-Grad-Gray-HSV	Data03	90.72
	Multi-score fusion	Gist-Grad-Gray-HSV-RGB	Data06	91.62
	Feature aggregation	Gabor-Gist-Gray-HSV	Data03	90.95
Linear SVM	Two-score fusion	Gist-Gray-HSV	Data01	**92.43**
	Multi-score fusion	Gabor-Gist-Grad-Gray-HSV	Data04	92.21
	Feature aggregation	Gabor-Gray	Data09	90.80
Ensemble Learning	Two-score fusion	Gabor-Gist-Grad-Gray-HSV	Data01	92.21
	Multi-score fusion	Gabor-Gist-Grad-Gray-HSV-RGB	Data06	91.92
	Feature aggregation	Gray-HSV	Data06	86.72
ANN	Two-score fusion	Gabor-Gist-Grad-Gray	Data02	92.06
	Multi-score fusion	Gist-Grad-Gray-HSV-RGB	Data05	91.69
	Feature aggregation	Gabor-Gray-HSV	Data01	89.54

Table 9: Sample Semantic Misclassification phrases

Phrase	TP	TN	FP	FN	Ratio
lift, hit, death, earthquake, magnitude, toll, least	1	1	1	1	1
hit, absolutely, damage, quake, reported, severe	1	1	1	1	1
hit, earthquake, quake, magnitude, coast	3	1	1	0	3
kill, earthquake, people, least	1	1	1	0	1
damage, storm, reported	1	2	2	1	0.5
bring, family, damage	1	2	2	0	0.5
bring, damage, storm	2	5	0	2	0.4

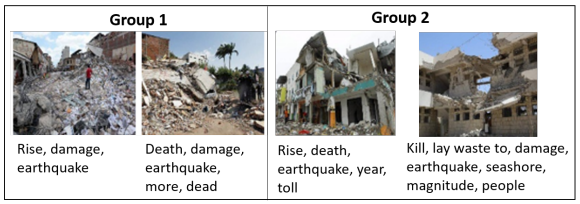

Figure 18: False Negatives

into two groups. Group 1 images represented *nature Damage* in an urban environment (houses and roads). The corresponding semantics had mutual words like "damage, storm, cause". Group 2 images mainly contained landslides which introduced sharp edges, a characteristic similar to *built-infrastructure damage*; the term "earthquake" was common in the corresponding annotations.

5.4.4 Level I vs. Level II Classification. Finally, we investigate how misclassified samples differed between Level I and Level II classifiers. Specifically, a Gaussian kernel SVM in Level I and a two-score linear SVM in Level II are compared. Table 10 indicates the number of FP and FN. Comparing Level I classifiers in rows 1,2 and 4 to Level II classifiers in rows 3 and 5, we notice that the number of FP decreases significantly (up to 5 times less) but the number of FN increases.

Figure 19: False Positives

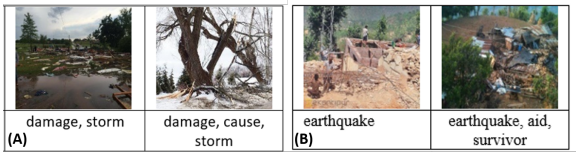

Figure 20: Misclassified Images: (A) FP and (B) FN

Figure 20 displays images that were misclassified by both Level II classifiers. Although no apparent visual indications justify this confusion, there may be a hidden feature that is shared among the classes, to which these observations are misclassified.

Considering the corresponding annotations, the first two rows in Table 11 correspond to the FN images. The phrase in the first row is correctly classified every time, while the term "earthquake" was not recognized as a TN since most of the corresponding observations were labeled as TP. For the FP, the observations were incorrectly classified after augmentation since their associated phrases were also misclassified, except for the two phrases in rows 3 and 4, which had no FP during semantic classification.

The misclassified images support the following conclusion: "damage" cannot be strictly identified visually. While this damage may be of one type, its classification is greatly influenced by the nature of the surrounding environment: fallen trees in urban environment and broken houses in rural areas were often misclassified. Moreover, damage can be relative, it is a fuzzy term. It can be used to describe a broken window, or a broken wall, just as confidently

Table 10: Level I vs. Level II Classification

Feature Set	Classifier	FP	FN
Data01	Gaussian SVM	205	22
Gabor-Gist-Gray-HSV	Gaussian SVM	105	21
Gabor-Gist-Gray-HSV-Data01	2-score linear SVM	25	77
Gray-Gist-HSV	Gaussian SVM	108	22
Gray-Gist-HSV-Data01	2-score linear SVM	25	77

Table 11: Annotations of Misclassified Images

Phrase	TP	TN	FP	FN
Aid, earthquake, survivor	3	0	0	0
Earthquake	90	11	11	0
Damage, cause, storm	1	6	0	1
Damage, storm	10	8	8	0
Damage, earthquake, cause, hit, magnitude, severe, storm	0	2	2	0
Aid, country, earthquake	0	1	1	0
Nothing	57	24	24	0

as describing fallen building or a chopped tree. With such a broad definition of damage, classification remains tricky, in the sense that features can't definitively capture "damage". For example, a fallen building exhibits vertical and horizontal edges similar to a fallen tree, both are damage, but distinct nevertheless. For this reason, multi-modal features improved the model of an image, and helped understand the type of damage present in it.

6 CONCLUSIONS

In this work, we proposed a computationally inexpensive multi-modal disaster-related categorization approach to classify Twitter data into *built-infrastructure damage* or *nature damage* classes. Manually-labeled tweets and attached images were represented using both semantic attributes and low-level visual features. All performance metrics (accuracy, precision, sensitivity, and recall) improved when using the multi-modal feature vector; accuracy improved to 92.24% when the probabilities of both feature modalities were concatenated to train a new Gaussian Kernel SVM. Future work will involve developing an unsupervised learning approach and performing sub-class understanding. This will eventually facilitate damage image indexing, retrieval and real-time disaster assessment for first responders.

ACKNOWLEDGMENTS

This work was supported by the National Council of Scientific Research in Lebanon (CNRS-L).

REFERENCES

[1] Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. 2012. Semantics+ filtering+ search= twitcident. exploring information in social web streams. In *Proc. 23rd Conf. Hypertext & social media*. 285–294.

[2] Firoj Alam, Muhammad Imran, and Ferda Ofli. 2017. Image4Act: Online Social Media Image Processing for Disaster Response. *IEEE/ACM Int. Conf. Advances in Social Networks Analysis & Mining*.

[3] Samar Alqhtani, Suhuai Luo, and Brian Regan. 2015. Fusing text and image for event detection in Twitter. *The Int. Journal of Multimedia & Its Applications* 7, 1 (2015), 27.

[4] Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining twitter to inform disaster response. In *Int. Conf. Information Systems for Crisis Response and Management (ISCRAM)*.

[5] W Ancy Breen, A Merry Ida, et al. 2016. Implementation of Speedy Emergency Alert using Tweet Analysis. *Indian Journal of Science and Technology* 9, 11 (2016).

[6] Cornelia Caragea, Nathan McNeese, Anuj Jaiswal, Greg Traylor, Hyun-Woo Kim, Prasenjit Mitra, Dinghao Wu, Andrea H Tapia, Lee Giles, Bernard J Jansen, et al. 2011. Classifying text messages for the haiti earthquake. In *Proc. 8th ISCRAM*.

[7] Gianluigi Ciocca, Claudio Cusano, and Raimondo Schettini. 2015. Image orientation detection using LBP-based features and logistic regression. *Multimedia Tools and Applications* 74, 9 (2015), 3013–3034.

[8] Koen Deschacht, Marie-Francine Moens, and Wouter Robeyns. 2007. Cross-media entity recognition in nearly parallel visual and textual documents. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*. Le Centre de Hautes Etude Internationales D'informatique Documentaire, 133–144.

[9] Centre for Research on the Epidemiology of Disasters. 2014. Annual Disaster Statistical Review 2013: The numbers and trends. http://reliefweb.int/report/world/annual-disaster-statistical-review-2013-numbers-and-trends

[10] Jim Gorman. 2006. 5 Natural Disasters Headed for the United States. http://www.popularmechanics.com/science/environment/a4875/3852052/

[11] Muhammad Imran and Carlos Castillo. 2015. Towards a data-driven approach to identify crisis-related topics in social media streams. In *Proc. of the 24th Int. Conf. on World Wide Web*. ACM, 1205–1210.

[12] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. Processing social media messages in mass emergency: A survey. *Comput. Surveys* 47, 4 (2015), 67.

[13] Muhammad Imran, Carlos Castillo, Jesse Lucas, Patrick Meier, and Jakob Rogstadius. 2014. Coordinating human and machine intelligence to classify microblog communications in crises. In *ISCRAM*.

[14] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. 2014. AIDR: Artificial intelligence for disaster response. In *Proc. of the 23rd Int. Conf. on World Wide Web*. ACM, 159–162.

[15] Hadi S Jomaa, Yara Rizk, and Mariette Awad. 2016. Semantic and Visual Cues for Humanitarian Computing of Natural Disaster Damage Images. In *12th Int. Conf. on Signal-Image Technology & Internet-Based Systems*. 404–411.

[16] Matt Mahoney. 2011. About the Test Data. mattmahoney.net/dc/textdata.html

[17] Stuart Middleton, Lee Middleton, and Stefano Modafferi. 2014. Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems* 29, 2 (2014), 9–17.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[19] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[20] Hussein Mouzannar, Yara Rizk, and Mariette Awad. 2018. Damage Identification in Social Media Posts using Multimodal Deep Learning. In *ISCRAM*. 529–543.

[21] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Ng. 2011. Multimodal deep learning. In *Proc. 28th ICML*. 689–696.

[22] Dat T Nguyen, Ferda Ofli, Muhammad Imran, and Prasenjit Mitra. 2017. Damage Assessment from Social Media Imagery Data During Disasters. *IEEE/ACM Int. Conf. Advances in Social Networks Analysis & Mining*.

[23] Soujanya Poria, Erik Cambria, Newton Howard, Guang-Bin Huang, and Amir Hussain. 2016. Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing* 174 (2016), 50–59.

[24] Mehran Sahami and Timothy D Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. of the 15th Int. Conf. on World Wide Web*. ACM, 377–386.

[25] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. 2010. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Systems, Man, & Cybernetics-Part A: Systems & Humans* 40, 1 (2010), 185–197.

[26] Nitish Srivastava and Ruslan Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *NIPS*. 2222–2230.

[27] Statista. 2017. Number of internet users worldwide from 2005 to 2017 (in millions). http://www.statista.com/statistics/273018/number-of-internet-users-worldwide

[28] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conf. Computer vision & pattern recognition (CVPR)*. 3485–3492.

[29] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *CVPR*. 8697–8710.