AMERICAN UNIVERSITY OF BEIRUT

HUMANITARIAN VISUAL AND SEMANTIC COMPUTING: CRISIS-RELATED IMAGE TWEETS CLASSIFICATION

by
HADI SAMER JOMAA

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
November 2016

AMERICAN UNIVERSITY OF BEIRUT

HUMANITARIAN VISUAL AND SEMANTIC COMPUTING: CRISIS-RELATED IMAGE TWEETS CLASSIFICATION

by

HADI SAMER JOMAA

Approved by:

Dr. Mariette Awad, Associate Professor — Advisor
Department of Electrical and Computer Engineering

Dr. Louay Bazzi, Associate Professor — Member of Committee
Department of Electrical and Computer Engineering

Dr. Daniel Asmar, Associate Professor — Member of Committee
Department of Mechanical Engineering

Dr. Carlos Castillo, — Member of Committee
Director of Research for Data Science at Eurecat

Date of thesis defense: November 11, 2016

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:

Jomaa	Hadi	Samer
Last	First	Middle

● Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☒ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after:

One ---- year from the date of submission of my thesis, dissertation, or project.
Two ---- years from the date of submission of my thesis, dissertation, or project.
Three ---- years from the date of submission of my thesis, dissertation, or project.

23/Nov/2016

Signature Date

ACKNOWLEDGMENTS

This thesis is dedicated to my family; my father Samer, my mother Hiam, my brother and my two sisters. I also like to thank my fiancée Najat Bdeir for showing me constant support.

Special thanks to my advisor and mentor throughout the pursuit of my Master's degree, Professor Mariette Awad for her motivation, constructive criticism, and for always pushing me to do better. This work has been done by collaborating with Professor Carlos Castillo, who shared his technical and work experience within the field of humanitarian computing, and helped me understand what this field needs.

My recognition and gratitude are addressed as well to Professor Daniel Asmar, who was the first person to introduce me to computer vision, giving me the opportunity to discover my passion, and to Professor Louay Bazzi.

AN ABSTRACT OF THE THESIS OF

Hadi Samer Jomaa for Master of Engineering
Major: Artificial Intelligence

Title: Humanitarian Visual and Semantic Computing: Crisis-Related Image Tweets Classification

When the public and first responders are flooding the internet with often annotated images and texts during natural disasters, rescue teams are overwhelmed to prioritize often scarce resources. Given that most of the efforts in such humanitarian situations rely heavily on human labor and input, we propose in this research a novel approach that leverages social media and uses machine learning and computer vision to help automate humanitarian intervention. After all, “an ANNOTATED image is worth a thousand words”. Our framework relies analyzing visually and semantically twitter data. We merge low-level visual features that extract color, shape and texture with semantic attributes extracted from annotated pictures posted on Twitter in disaster times. These visual and textual features are trained and tested on a home-grown dataset solely gathered from Twitter. The best accuracy obtained after crowdsource labeling the data was when low-level visual features and semantic attributes were applied to the different classification scenarios of support vector machines (SVM), Neural Networks and Ensemble Learning-SVM with 5-Fold cross-validation. Since the data is organically unsupervised, we proposed a structural neuronal modification to the cortical algorithms, a deep learning algorithm inspired by the human visual cortex and tested it using the previously proposed visual and semantic features. As expected, the proposed CA showed better performance than regular CA, which motivates follow on research.

CONTENT

ILLUSTRATIONS

TABLES

CHAPTER I

INTRODUCTION

A Natural Disasters and Crisis

We live in a harsh world. Technocratic plates that hold earth's crust are ever shifting causing earthquakes, volcanos and tsunamis. Man caused pollution over the past century had its toll on the global climate, leading to global warming. This change in weather, as modeled by NASA, includes the risk of drought, floods, and increased intensity of storms making hurricanes more intense (Riebeek 6). In the Annual Disaster Statistical Review published in 2013 (Debarati, Hoyois and Below), the average amount of natural disasters triggered over the span of a decade from 2003 till 2012 is estimated as 388 annually with an estimated average cost of economic damage of 156.7 billion US $. Looking ahead, the United States is up against 5 major disasters like nothing that has ever passed. A geologist commenting on one of the expected events says that: "There's no guarantee there would be any advance warning" (Gorman 1). The case holds for almost all disasters. However, environmental organizations around the world don't just wait for the imminent to occur. The United Nations, Red Cross Red Crescent, Engineers without borders, are examples of the leading organizations that work on disaster relief. UNDAC, United Nations Disaster Assessment and Coordination, an international response system, is placed on standby and can be deployed to any country within 48 hours ("United Nations Disaster Assessment and Coordination."). UNISDR, United Nations International Strategy for Disaster Reduction, is also part of the UN Secretariat and notably defined to serve as a system that coordinates disaster reduction activities of UN systems and regional organizations ("The United Nations Office for Disaster Risk Reduction.").

In addition to this world being harsh, it is war-torn as well. Out of the 162 countries recognized by the Institute for Economics and Peace, only 11 countries were not involved in any kind of conflict (Withnall). Wars and riots cause immeasurable damage that needs to be dealt with as well. In the recent riot that broke down in Baltimore, an estimated 9 million USD in just 8 days (Toppa). In December 2010, the “Arab Spring” broke down and regimes started falling apart. Demonstrations and protests were organized, leading to tension with the governments, fights with law enforcement, and structural damage. This phenomena spreads over several Arabic countries starting with Tunisia, going through Egypt, Sudan, Yemen, Bahrain, Libya, and Morocco. According to an article published by Reuters News, HSBC bank estimated the total damage worth 800 billion USD ("Arab Spring to cost Middle East $800 bln, HSBC estimates."1).

Disasters' damage, be it a result of Nature or Man, hence can't be prevented, but can be dealt with and anticipated intelligently, benefitting from recent advancement in technology, and viral growth of social networking.

B Humanitarian Engineering and Computing

Humanitarian Engineering and Computing in general is the branch of engineering and science that focuses on issues which limit opportunities and development in communities. It is also not necessarily restricted to being a reaction to a disaster or a crisis ("Humanitarian Engineering and Computing."), however in this thesis; we highlight disaster related solutions. Numerous engineering applications and solutions have been sought to reduce disaster effects. At UMass College of Engineering ("Engineering for Everyone Tackles Disaster Relief."1), a sustainable earthquake-resistant hospital developed by team of students. Seven-foot dampers were introduced in the design at every floor to absorb shock waves, along with horizontally flexible material to decouple the structure from the ground. The Aid Necessities Transporter aka A.N.T., developed by designer Bryan Lee serves the purpose of rapidly providing damaged communities with supply pods and temporary shelter (Banks). Taking advantage of solar energy, LuminAID was created by a couple of students at Columbia University ("LuminAid."). An inflatable waterproof LED, which provides light for up to three years, can serve as a light source and a beacon in damaged areas where no electricity as accessible. These and many more solutions can be labeled under Humanitarian Engineering.

However, we exist in a digital world where anyone with internet connection can tap into virtually anywhere and know virtually anything. Information easily travels across continents, and knowledge is transferred among millions at viral rates. If the saying:" Two minds are better than one" is true, then the internet, where millions are connected is the best place to find solutions. Social media aids in alerting people and spreading awareness, locating missing people, gathering donations, forming volunteer groups, organizing public movements. We are beyond the times where the sole source of information was the television and the radio. Governments also can build on social media's presence. Bruce R. Lindsay, an analyst in American National Government notes that "if FEMA adopted social media use for recovery, the agency could provide information concerning what types of individual assistance is available to individuals and households, including how to apply for assistance …" (Lindsay). Smartphone applications also exist for the purpose of disaster management and relief. Some of them include EarShot, which gathers eyewitness reports from the scene and helps in submitting service reports. Shelter View is an application that alerts people to the nearest disaster facility. FEMA is the official app for the Federal Emergency

Management Agency, which include kits and checklists, along with other procedures in case of an emergency. In short, the ever growing audience of social media users such as Twitter, Facebook, Snapchat, etc…, have provided a huge online database of texts and images readily available that can be directed towards developing humanitarian applications (Castillo). A twitter-based humanitarian application was recently developed and can be considered at the core of humanitarian computing.

Humanitarian computing also spans over a large spectrum: it includes applications that spread awareness and alerts about possible natural disasters. It also includes information-processing methods that extract actionable information from social media which aid in reducing disaster response time, such as the one proposed by (Imran et al.) and currently implemented on Artificial Intelligence for Disaster Response (AIDR) (Imran et al. 159-162). AIDR is an important application that can handle overwhelming data transferred during a crisis, and results in an automatic real-time classification of tweets. However, an image says a thousand words, which means that images convey the required information clearer than any description. Depending on a text alone in describing infrastructure damage, poverty, droughts, floods, etc… is not sufficient. That is why image processing techniques and computer vision must be integrated in such platforms.

The existing approaches ignore the images often attached with these messages, which may contain even more valuable information than words. Strictly speaking, no humanitarian computing applications were developed that leverage annotated images that float abundantly on social media, where they solely focused on NLP and human effort.

C Motivation

Damage caused by natural and man-made phenomena lead to loss of lives, destruction of crops, floods, infrastructure dysfunction, and more. In order to mitigate casualties resulting from the damages, several governmental and non-governmental organizations around the world dispatch response teams and resources to places where the disasters strike. This after-the-fact response can be slow and sometimes misdirected as it relies on generally inaccurate information obtained by these organizations.

We also live in a world that is virtually connected through social media websites scattered all over the internet: around 3.174 billion people were reported online in 2015 by ("Internet Users."). Through social media applications and websites, images and messages travel as fast as a click of a button. By making use of communicated images, dedicated man power can be relieved from the tedious task of identifying the damage type and severity caused in different places by natural disasters. Instead, their focus can be redirected towards better management of the help and rescue missions by initiating a better response and prioritized dispatch of units. It may also make the response quicker and more efficient, since in times of crisis, infrastructure damage and overwhelming use of telecommunication on the networks impedes proper communication as the region grows into a state of chaos and panic. Images can be used as a form of validation, making reports more descriptive and conveying a more powerful message than words alone. This can be achieved by employing advanced image processing analysis and scene understanding techniques to automatically categorize and assess images containing depictions of damage. Over the past couple of decades, scene understanding has been applied to identify moving objects in an image, identifying visual attributes, enhancing segmentation, boundary detection, recognizing visual phrases and several other objectives. Despite all the advancements in computer vision, no single algorithm can describe an image on all semantic levels. For every set of categories, a separate classifier can be trained. This leaves room for innovation and contribution by creating classifiers for semantic categories and visual attributes that have not yet been considered.

D Proposal

In this thesis, we propose a novel hybrid approach that merges semantic attributes extracted from disaster-related messages with low-level features extracted from the corresponding images to aid in humanitarian computing. While some of the crisis-related topics suggested in (Imran and Castillo 1205-1210), such as “Donations and Volunteering” or “Caution and Advice” may or may not typically include images that can be properly exploited, topics such as “Infrastructure and Utilities” are more commonly associated with images from which we can assess the existing type of natural disaster damage. For this reason, we investigate the messages describing damage in this thesis, and divide them into two classes: built-infrastructure and nature damage. (Mileti) states that disaster losses, damage, are the result of interactions between the physical environment, social demographics and the built environment. The losses to the physical environment, as a result of hazardous events, is termed as nature damage, while the losses to the built environment which includes buildings, bridges, roads, etc. is referred to as built-infrastructure damage. The approach suggested in this paper is a two-step hybrid classification scheme. At the first step, visual features and semantic features are trained on separate classifiers. The scores of the outcome from semantic and low-level classification are aggregated as a new feature vector, hybrid, which is further trained and tested on a new classifier, two-step. An unsupervised clustering approach is also proposed, by modifying the regular CA to become more biologically plausible. The architecture is changed, by introducing recurrent connections, leading to more complex activity and hence higher spikes. These unsupervised approaches, CA and the enhanced, Strong CA are tested on the visual features that yielded the best supervised classification results, and compared with raw pixel data. The proposed semantic descriptors are also clustered and compared with the clustering of descriptors resulting from raw semantic processing.

E Contribution

The key contributions in this paper are summarized as follows. We propose a new class for image classification, types damage, which is, to the best of our knowledge, unprecedented in the field of visual humanitarian computing. We validate the authenticity of our work by creating a data set gathered solely from Twitter based on keywords related to ‘damage’, in which the observations are presented as images and their corresponding annotations. We demonstrate how a bag of words can be utilized to create the semantic features, by processing the reoccurring words in the database as well as their synsets (WordNet (Miller et al.)), in an iterative form to achieve coherent set of words. Representing an image using visual features in an important step towards proper learning and classification. We explore a set of inexpensive low-level visual features that convey distinct aspects of an image, and assert that by combining these features, classification improves. The proposed classification scheme relies on a new feature vector that is made up from the score of the outcome achieved from semantic and visual classifiers independently trained. Finally, a modified cortical algorithm network is created for image and annotation clustering, based on the work done in (Hajj, Rizk and Awad 327-334). The proposed *Strong CA*, is introduces recurrent connections for stronger spike activity as well as a more anatomically accurate model of the neuron.

Throughout the rest of the thesis, we go into the technical details behind the proposed approach in the Methodology Chapter, were we address the low-level features used, the proposed semantic processing scheme which includes the bag-of-words formation and the semantic descriptor processing, and the different learning approaches. In the Database Chapter, we demonstrate how we developed the dataset based solely on queried tweets and manual classification. The Experiments Chapter includes all the results from the different learning approaches, were we do error analysis on the classifier that achieved the best accuracy. Finally, a summary of the proposed approach which highlights the results and the contribution of this thesis is presented in the Conclusion Chapter.

CHAPTER II

RELATED WORK

In our work, we represent an image via a combination of features that convey different information about the image, i.e., shape, color, texture, and energy. The annotations corresponding to the images are projected on a bag of words which results in a binary semantic descriptor. The visual and the semantic features are trained separately on different classifiers. The score of the decisions at the first stage are aggregated together as a new feature vector in used to train the classifiers at the second stage. This two-step hybrid approach asserts that by making use of image annotations, the classification $Accuracy$ is enhanced. This chapter describes some of the existing scene understanding approaches and their applications. Although the presented methods don't directly relate to humanitarian computing, it is critical to understand the steps behind image classification before developing humanitarian-oriented algorithms. Strictly speaking, to the best of our knowledge, so far no humanitarian computing applications have been developed that leverage annotated images found abundantly on social media.

A Natural Language Processing

Natural Language Processing (NLP) encompasses a wide range of fields in literature, such as auto summarization, parsing, sentiment analysis, word disambiguation, etc. However, in this section, we look into some of recent NLP techniques adopted for short text classification, since tweets are limited to 140 characters, as well as a look on some applications developed for humanitarian computing purposes.

1. Short Text Classification

(Sahami and Heilman 377-386)classify documents based on a Kernel approach which compares the similarity of short text snippets. Snippets are utilized as a query for a search engine and then by computing the TFIDF (Term Frequency–Inverse Document Frequency), are used to model the document. (Yih and Meek 1489-1494) use the relevance weighted inner-product of term occurrences to model the document. (Zelikovitz and Hirsh 1183-1190) develop an approach that relies on WHIRL (Liu and Singh 211-226), a tool that can search and retrieve text information under specific conditions, to find short text similarity. (Phan, Nguyen and Horiguchi 91-100) employ external information gathered from universal dataset to classify short text. The dataset is analyzed via Latent Dirichlet Allocation (LDA), followed by Gibbs sampling for topic inference. Classification is done by a Maximum Entropy classifier. (Chen, Jin and Shen

1776-1781) predefine the topics and using a multi-granularity topics space approach, and train a regular SVM classifier. Short text classification is also useful in web page classification, as suggested in (Charalampopoulos and Anagnostopoulos 235-239), who used Weka tool (Holmes, Donkin and Witten 357-361) for clustering and classification of synthesized data representing web documents.

2. NLP-Based Humanitarian Computing applications

Humanitarian computing approaches so far have been only addressed through the use of natural language processing (NLP) as well as human effort (Imran et al. 67; Imran et al. ; Cresci et al. 1195-1200). (Imran et al. 67), detail in this survey the existing approaches that deal with natural disaster responses through the use of social media. Some of these systems do not particularly address the needs of emergency responders, but are framed as a way to process social media information. (Imran et al.), Imran et al develop a platform, *AIDR*, that uses the manual effort provided by microblog streams as the training set for classification of crisis-related messages. A similar platform, called *Crisis Mapping* (Okolloh 65-70)which employs digital volunteers also emerged to help collect, classify and geotag information. This tool was first developed as a website platform were people would send out an SMS to a local number, and the text then synchronizes with the platform before showing up on the website. This tool later made reporting on a disaster easier by facilitating reports through mobile phones with internet accessibility. *Crisis Mapping*, as the name suggest, presents users with a map displaying possible crisis based on public reports. However, it is not automated. *Twitcident* (Abel et al. 285-294) is another web-based platform which aims to extract relevant information from regarding incidents from social web streams, particularly Twitter. The proposed framework detects incidents from emergency broadcasting services and starts to profile the incident based on several variables including the time and type of the reported incident. The profiled incident undergoes then semantic enrichment. This step includes gathering messages from the streaming API of Twitter, followed by name entity recognition, classification based on hand-crafted rules, calculates a facet-value pair from the links in the tweets and finally extracts meta-data regarding the tweet itself. Observations that are not related to the incident are filtered out using a thresholding formula that depends on the enriched tweet and the facet-value pairs. *Twitris* (Purohit and Sheth) is an analytics platform that is constantly collecting tweets. These tweets are then analyzed in three different manners, Spatio-temporal-thematic in which observations are clustered based on the location, time and topic of the tweet; People-Context-Network analysis that identifies contextually important people to be addressed in the community; Sentiment-Emotion-Subjectivity analysis where the emotion presented in the tweet is investigated towards understanding an event-oriented community. *Tweedr* (Ashktorab et al.) on the other hand is not a web-based platform, but rather a Twitter-mining tool that aims to extract actionable information from tweets for disaster relief. Tweets are represented in as a standard unigram feature vector and several classifiers are tested including KNN and logistic regression. Extraction is done on the positive examples, using a conditional random field trained on five different disaster datasets. *EMERSE* (Caragea et al.) is an iPhone application that collects and classifies tweets and was developed to the Haiti earthquake response. The designed Twitter crawler gathers the most relevant tweets for specific keywords. Each observation is first represented by a feature vector using the bag-of-words approach. Features are then filtered using feature abstraction, in order to reduce the feature dimension and get rid of redundant information, feature selection.

B Scene Understanding

In general, scene modeling approaches may be divided into two categories. The first is based on traditional pipeline of object detection; where features include appearance, motion, tracking of the detections. The second uses appearance features directly and requires supervised training. Features and approaches vary across methods, and depend highly on the application. Following is an overview of some scene understanding techniques categorized into 3 groups. In Figure 1, we present an overview chart which depicts the types of classifiers used vs the type of classification investigated.

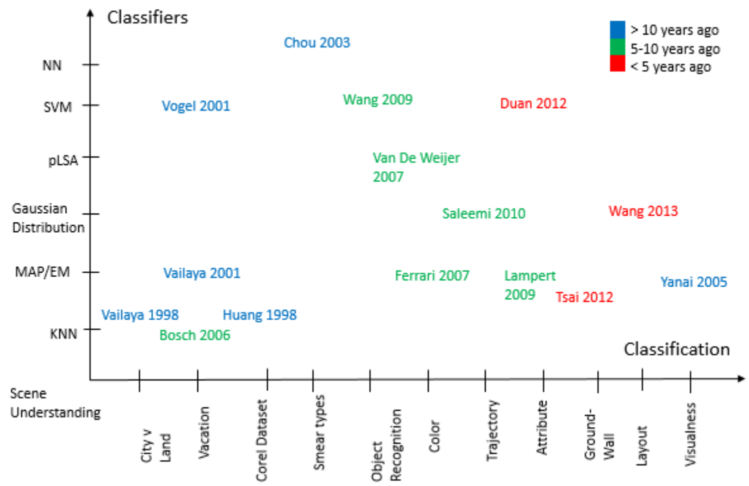


Figure 1 Scene Understanding Related Work Overview

1. Hierarchical Classification

(Vailaya, Jain and Zhang 1921-1935) tackles City/Landscape image classification through developing a hierarchical classifier that utilizes KNN. Based on a proposed feature saliency method, they resort to edge direction coherence vector for City/Landscape separation and color histogram to separate natural images. In (Vailaya et al. 117-130), Vailaya further extends his work to photos taken by users on holidays. This problem is addressed by modeling the class-conditional probability density functions based on Bayes' theory and classification is accomplished via MAP estimation. Features used include HSV and LUV color information along with MSAR texture features (Mao and Jain 173-188). A modified MDL criterion is implemented to reduce the dimensionality of the feature space. (Huang, Kumar and Zabih 219-228)use banded color correlograms as features, which describe how the spatial correlation of colors changes with distances. The RGB color space is quantized into 512 colors and singular value decomposition is applied to reduce/eliminate the noise. Classification is done using the nearest-neighbor method (with the closest one dropped). Then normalized cuts are applied to split the sub-classes in a way that maximized intra-class association and minimize inter-class association. The proposed approach was tested on 11 classes from the Corel collections and showed that Correlograms performed better than regular RGB features. (Chou and Shapiro 150-168) use clustering as an intermediate step for feature processing before classification. Component classifiers are trained to learn the subclasses (clusters), which serve as a non-linear kernel on the original features. The combined outputs are then trained on a neural network for pap smear prescreening. (Bosch, Zisserman and Muñoz 517-530)use probabilistic latent semantic analysis to fit the training images. Four dense descriptors are investigated where the images are concatenated into visual words created by K-means clustering. Topics are learned by fitting a pLSA model. The document specific mixing coefficients are computed and classification is accomplished through KNN. (Vogel and Schiele 133-157) incorporate the 9 local semantic descriptions proposed in (Mojsilović, Gomes and

Rogowitz 79-107) to recognize natural images. SVM concept classifiers are trained on the sub-blocks of the images, and the outputs are augmented in a concept-occurrence vector. Extracted features included HSI color histogram, edge direction histogram, and the 24-features of the gray level matrix in four principal directions; contrast, energy, entropy, homogeneity, correlation and inverse difference moments. SVM was used for training and compared with the Prototype approach.

2. Learning through Attributes

(Wang and Forsyth 537-544)learn to locate an object/attribute pair. Corresponding saliency maps are generated for every image, and classification is done by a multiple-instance SVM over window sets in the training images, where every window gets an average value depending on the pixel saliency value. Homogeneity is included to detect objet boundaries. This results in two optimization problems coupled by instance labels. Joint and separate learning are tested using features that include color, coarse histogram oriented gradients, HOG (Dalal and Triggs 886-893), and texton-features. (Van de Weijer, Schmid and Verbeek 1-8) assign color names to regions from real-world images. Images are collected in the sRGB format and are gamma corrected. The images are changed to the LAB space, and pixel values are assigned to location on a 3D grid. They train a pLSA to recognize 11 basic colors. The method is evaluated on three datasets gathered from Google, Ebay and the Chip-based color name set. (Ferrari and Zisserman 433-440)that learn colors and patterns of segments by learning likelihoods of attributes of the labeled images. The basic units for the generative model are the segments extracted by the approach devised in (Felzenszwalb and Huttenlocher 167-181), which undergo a preprocessing phase prior to learning. Pixels are assigned to patch types, and their types are aggregated in a histogram, normalized over the segments. Segment histograms are clustered to form an appearance codebook, and segments are assigned to the nearest bin. Geometric properties related to pairs of segments are also investigated. In the training phase, the location of the attribute is not provided, only its existence. Learning is done iteratively and exhaustively. (Lampert, Nickisch and Harmeling 951-958) introduce attribute-based classification, where attributes are learned through low-level features, and are then able to predict unknown classes. The problem is formulated as learning with disjoint training and test classes. Two attribute-based classifiers are devised, direct attribute prediction (DAP) where a layer of attribute variables is added to decouple images from the layer of labels, and indirect attribute prediction (IAP), where the attributes form a connecting layer between known classes of the training phase and unknown classes of the test phase. (Yanai and Barnard 419-422) study the visualness of attributes, by calculating entropy and locate attributes using EM. Image features, i.e. color, texture and shape are extracted from images labeled with concepts (X). Independent Gaussian mixture models are learnt via EM on equal numbers of segments from X images and non-X images. Two models are formed, one for the concept and the other for the non-concept and applied to the image segment. This iterative method incorporates regions to enhance the X-model and probability value. Knowing the region of the concept allows the calculation of the entropy. (Torresani, Szummer and Fitzgibbon 776-789) develop a new descriptor for object recognition, through training classifiers on known categories, and concatenating real-valued outputs of these base classifiers, called classemes, when applied to a new image to form the new descriptor. The approach presented is divided into classeme learning and any object-related category learning. (Berg, Berg and Shih 663-676) used MILBoost to generate a probability map which locates the specified attribute. Potential attributes are collected based on occurrence frequency in the text describing the images, and distributed into separate datasets. Visualness is computed based on the average labeling precision of the respective classifier. To prevent the use of redundant attributes, a visual synset is built that contains synonyms of the attributes, and classifiers with significant mutual information are joined. Visual features are created using SIFT, HSV, and texture descriptors.

3. Probabilistic Approaches

(Saleemi, Hartung and Shah 2069-2076)focus on statistically modeling trajectories obtained from surveillance footage a hierarchical unsupervised approach.

The optical flow, modeled through pixel location (2-elements), magnitude and direction of the optical flow, is taken as a variable and modeled in a probability function of motion patterns. The 4D distribution is modeled as a mixture of Gaussian components, and observations of the optical flows are clustered in each dimension, to compute the likelihood of pixel and optical flow being part of a motion pattern. (Tsai et al. 121-128) present a real-time approach that locates ground-wall boundaries. This is done by utilizing motion cues to compute likelihoods of hypotheses. These hypotheses are modeled as a Bayesian posterior probability distribution which is identified and updated from the video information. This geometric method is applied to get 3D structural data about the image, governed by surrounding assumptions a set of logical constraints. KLT (Shi and Tomasi 593-600) is employed to track point features across frames. The likelihood for every hypothesis is computed at every frame, and by applying a Bayesian filter, the likelihoods are aggregated across the frames. (Wang, Gould and Roller 92-99) retrieve room geometry and furniture layout. The inconsistency between decorations and layout is addressed by adding latent variables to account for clutter. Inference is done without hand-labelling the clutter. Three vanishing points are detected in indoor scenes, whereas long lines are detected and clustered into three dominant groups. Super-pixels are utilized after over-segmentations and are added to the vanishing points to form the appearance model. The box layout is estimated by fitting a multivariate

Gaussian with diagonal covariance training. (Duan et al. 3474-3481) investigate local attributes that contain both discriminative and semantically meaningful value in the image. Images are annotated by a class label, corresponding to the species of the animal and an object bounding box, whereas attribute name and location are unknown. This task can be configured as an inference problem on a latent conditional random field. Finding a region containing the attribute is formulated as minimization of an energy function containing the preference of a classifier and pairwise relationships. An SVM is trained on latent regions from both positive and negative images. Resulting weights are utilized in the energy function.

C Feature Fusion

Fusing textual and visual clues to boost the performance of classification is a concept used often in machine learning. (Deschacht, Moens and Robeyns 133-144) employ WordNet (Miller et al.) to form synsets of salient words in annotations. Based on a salience measure and measure of visualness inspired by (Kamps et al.), a probabilistic algorithm that asserts the present of the entities within the image. (Feng and Lapata 272-280) train a continuous relevance image annotation model, which learns the join probability distribution of words and image regions. The words' presence of absence is obtained from a Multiple-Bernoulli distribution that takes into account not only the associated annotations of the image, but rather the whole document for context. (Alqhtani, Luo and Regan)aggregate semantic descriptors, created from bag of words using TFIDF, with a set of texture and color visual features. They use KNN for classification, and prove that fusing these features enhances the performance. (Poria et al. 50-59)go one step further by fusing audio features with both visual and textual information for multimodal sentiment analysis. To get effective results, the information extracted is merged on a feature-level and on decision-level. In an attempt to improve image clustering, (Ma et al. 1555-1557)model the observations by adding two semantic features, one collected from the documents gathered from Wikipedia, Wisdom of Crowds documents, and the other on the text documents that form the corpus, and visual features extracted from the image in matrix format. Non-negative matrix factorization is employed for clustering. Latent Dirichlet Allocation is then employed to model the topics of each document.

CHAPTER III

METHODOLOGY

In this paper we suggest a two-step hybrid approach that utilizes the decision and score from low-level feature classification extracted from images, as well as the decision and score of semantic descriptor classification, extracted from their corresponding annotations, and trains the new feature vector to classify the types of damage resulting from natural disasters. Visual and semantic feature aggregation is also investigated. Representing an image via proper low-level features is one of the most challenging steps in scene classification. In the following we define the suggested semantic attribute descriptor, which is created by projecting the description on a bag of words, as well as some common color, shape, and texture features used in scene understanding. These low-level and semantic features are trained independently on an SVM classifier with linear and Gaussian kernel, using cross validation to choose the parameters, as well as Ensemble learning and Neural Networks, for comparison. The workflow of the proposed methodology is presented in Figure 2, while the off-page references are shown in Figures 11, 12 and 13. An unsupervised clustering algorithm is also described. *Strong CA* is a modified version of the regular CA where we change the model of the basic computational unit, the neuron, and introduce recurrent connections rendering the more human-brain-like. Unlike regular CA which is employed for supervised learning, Strong CA deals with the output from the architecture as a new representation, subsequently used for clustering. Unsupervised clustering is tested on the visual features that achieved the best accuracy and its results are compared with clustering of raw pixel data. The proposed semantic descriptor is also clustered and its results compared with those obtained from clustering descriptors from raw semantic processing.

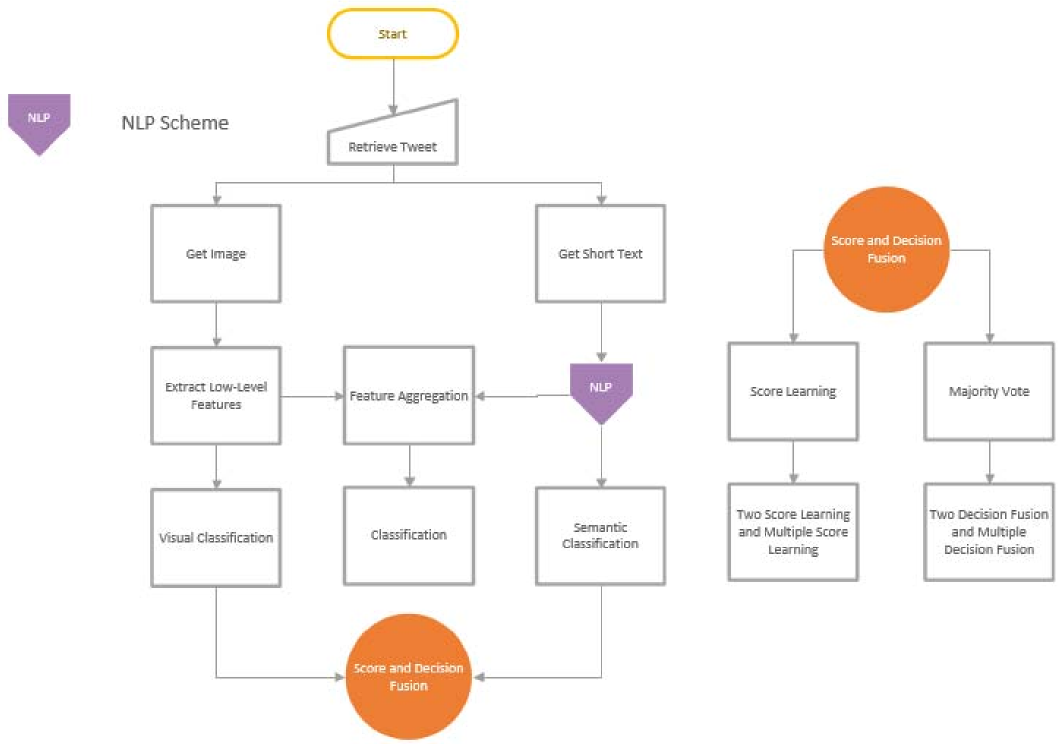


Figure 2 Workflow of the Approach

A Low-level Features

Table 1 Feature Information

Feature	Type	Vector Size	Time
RGB Color Histogram	Color	768	0.148
HIS Histogram	Color	84	0.135
Histogram of Image Gradient Directions	Shape	720	1.104
Gray Level Co-Occurrence Matrix	Texture	168	0.448
Gabor	Energy	30	0.253
GIST	Energy	512	0.437

The features in Table 1 are chosen to encompass distinct aspects of the image, specifically color through the Red-Green-Blue (RGB) histogram, and Hue-Saturation-Intensity (HSI) histogram, shape (Histogram of Image Gradient Direction), texture (Gray Co-Occurrence Matrix, GLCM), and energy through GIST and Gabor feature vectors. In addition, these features are computationally inexpensive, dense, and well used in the field of scene understanding. The features are extracted from 256×256 images using MATLAB R2015b on a computer with Intel(R) Core(TM) i7-4700MQ and 2.4GHz CPU. The time is in milliseconds.

1. Color

a. RGB Color Histogram

The 1-Dimensional histogram is based on the R, G and B color channels. The bin sizes can vary, however we the regular 256 bins per channel, and aggregated the features together. An example of the different color channels is presented in Figure 3.

Figure 3 RGB Color Channels Example

b. <u>Hue-Saturation-Intensity</u>

The Hue-Saturation-Intensity (HSI), used interchangeably with HSV, feature vector is an 84-bin histogram, 36 for hue, 32 for saturation and 16 for intensity. Hue and saturation are scale and shift-invariant with respect to light intensity. Figure 4 represents the HSI color channels.

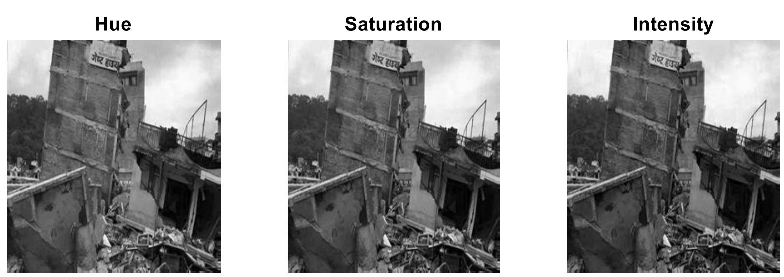


Figure 4 HIS Color Channels Example

2. *Shape Features*

a. <u>Histogram of Image Gradient Directions</u>

Based on the assumption that urban scenery contains more edges than natural images, vertical and horizontal that may form due to the presence of objects such as buildings and so forth, it could be of benefit to study the distribution of gradient direction of pixels in the image. The gradient histogram is divided into variable bin size. In the proposed approach, we set the number of bin sizes to 720, which corresponds to gradients of 0.5 degrees of resolution, to capture the fine gradients in the image. In Figure 5, 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.

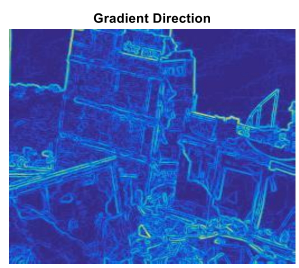


Figure 5 Gradient Directions Example

3. Texture Features

a. <u>Gray-Level Co-Occurrence Matrix</u>

GLCM, is a statistical method that calculates information about the texture of the image using spatial relationships between pairs of gray-value intensity pixels. The statistics include correlation, contrast, homogeneity, energy and entropy for specified displacements, i.e. offsets, in the image, which dictate the nature of the spatial relationships between every pair of pixels. Figure 6 shows the value of the feature vector on every pair of pixels at specified offsets.

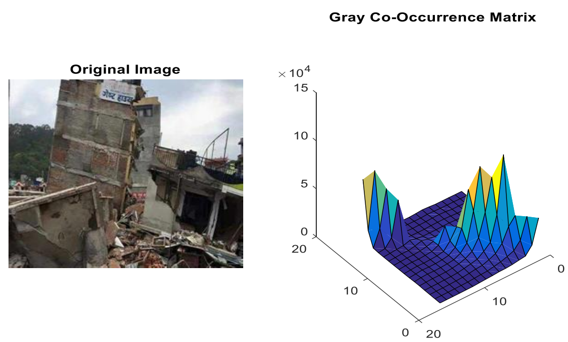


Figure 6 Gray Level Co-Occurrence Matrix Example

4. Energy Features

a. <u>Gabor</u>

The Gabor feature vector captures the energy of the image using FFT and gets the response of an image at various scales and orientations. This helps capture object patterns and edges at varying frequencies and orientations, such as Figure 7.

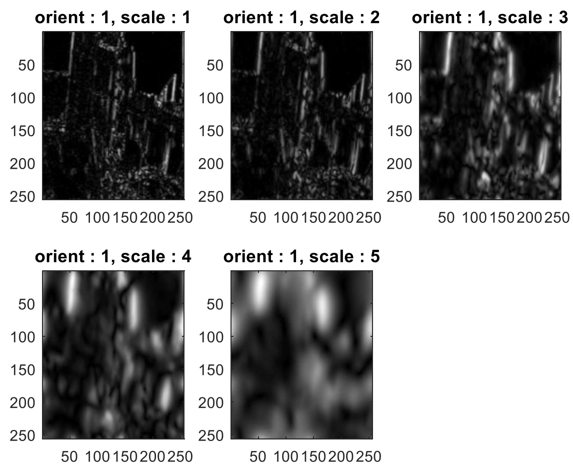


Figure 7 Gabor Feature Example

b. <u>GIST</u>

GIST (Oliva and Torralba 145-175)summarizes the gradient information (scales and orientations) for different parts of an image, which provides a rough description (the gist) of the scene. The description is based on a set of perpetual dimensions, mainly naturalness, openness, roughness, expansion and ruggedness. Figure 8 represents the gist of the image at different parameters.

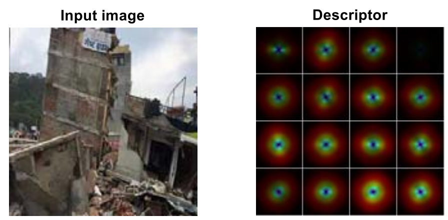


Figure 8 GIST Feature Example

5. Feature Combinations

In order to study the effect of multiple features on classification, different subsets of features are combined together, by concatenating the corresponding normalized feature vector into one. The features are primarily tested individually and used as a baseline for comparison against augmented features. The results of this step are addressed in Experiments Section.

B Semantic Descriptors

In this subsection, the steps that lead to creating the bag of words, which serves as basis for the semantic descriptors, are detailed. The formation and processing of the semantic descriptors are also addressed.

1. Bag of Words

a. Obtaining the Synonyms

In the established dataset, every entry (tweet) is associated with a set of words as provided by Twitter users. The first step is creating a list of the distinct words, referred to as *Terms*, available to establish a certain background on the words mostly used in describing damage images. Some of the most frequent *Terms* are shown in Table 2. An intermediate filtering step is used to get rid of redundant words, leaving only the distinct terms. The synonyms of these distinct *Terms* are gathered via WordNet (Miller et al.), a large lexical database of English. Nouns, verbs, adjectives and adverbs which are grouped into sets of cognitive synonyms (*Synsets*), each expressing a distinct concept, similar to (Wang and Domeniconi 713-721), (Hu et al. 179-186) and (Elberrichi, Rahmoun and Bentaallah 16-24). *Synsets* are interlinked by means of conceptual-semantic and lexical relations. The main relation among words in WordNet is synonymy, as between the words shut and close or car and automobile. However, the most frequently encoded relation among *Synsets* is lexical relations, super-subordinate relations, meronymy, and antonymy. Hence, the number of *Synsets* to which a words belongs to differs from one to another. In total, 50 *Synsets* were studied. We stopped at 50 since no change in the *Synsets* occurred between levels 49 and 50 based on the words in our dataset. Throughout the rest of this paper, semantic levels and *Synsets* are used interchangeably. Identifying related distinct terms is the next objective, and hence the synonyms of the distinct terms that occur at least 3 times are gathered on the corresponding available semantic levels. The result is a set of *Terms* with their respective synonyms at different levels such as those presented in Table 2.

Table 2 Sample Base Words and Synonyms

Terms	Frequency	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

b. <u>Creating the Bag of Words</u>

At every semantic level, a bag of words is created through the following steps. First the synonyms of every *Term* are compared to all other synonyms belonging to the other *Terms*. When a mutual synonym(s) is found, for example the word '*temblor*', in common between *Terms* '*earthquake'* and '*quake*', both groups of words (the *Terms* and their synonyms) are concatenated together and filtered from redundant synonyms, which in this example results in the *Term* '*earthquake*' having '*quake, temblor, and seism*' as synonyms, and the *Term 'quake'* removed from the list of *Terms*. After every combination, new mutual synonyms may emerge, hence this step is repeated for every *Term* until no mutual synonyms exist between the *Terms*. Every entry of the bag of words ends up containing a set of *Terms* and their synonyms that are related to each other through at least one synonym. An example is shown in Figure 9.

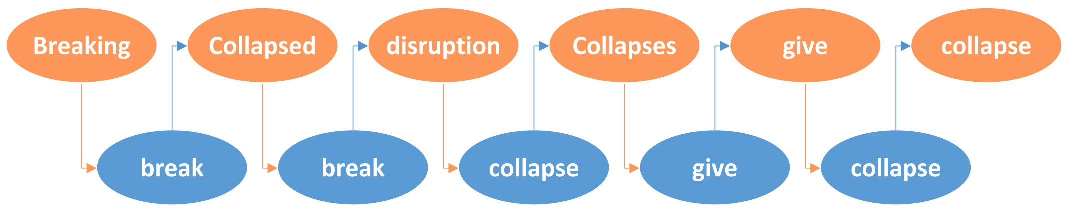


Figure 9 Combined Words (orange) and Joining Words (blue)

This however, leads to *Terms* connected to each other through unrelated synonyms. Considering Figure 9, the *Term* '*give*' is not highly related to '*breaking*' but still connected to it since a synonym of the word '*give*' may be present in the synonyms of the word '*breaking'*. In order to keep the list of combined *Terms* diverse, yet relevant, an additional step is devised. The synonyms that connect the *Terms*, called '*Joining Words*', are explored. Every set of combined *Terms* that shares at least 1 synonym are grouped together, and separated from the set previously established, this leads to new *Terms* and new synonyms. The set of *Combined Words* in Figure 9, is hence divided into 2 distinct sets. The first set corresponds to the *Term 'breaking'* and its *Combined Words'* synonyms, presented in Figure 10, whereas the *Term 'give'* is excluded from both the *Combined Words* and its synonyms removed from the separate set.

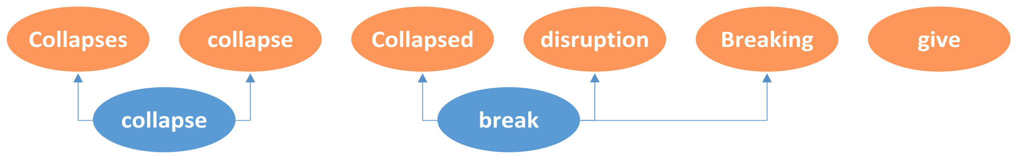


Figure 10 Filtered Combined Words

Some of the resulting terms and their corresponding group of words are shown in Figure 11.

Figure 11 Sample Terms from Data01

2. *Semantic Descriptor*

a. <u>Forming the Semantic Descriptor</u>

After the Bag of Words are created, it is possible to create the semantic descriptors. As previously mentioned, every entry in the data set consists of an image and annotations. These annotations are projected on the bag of words. In other words, every word in the annotation is looked up in the bag of words, and replaced by the lead root, to which every word is related. This transforms the annotations into combinations of lead roots, instead of different (related) words, referred to as filtered annotations.

The semantic descriptor is a binary descriptor, with a vector size equal to the length of the bag of words. Each feature corresponds to a lead root, and is flagged if the root is present in the filtered annotation. One of the drawbacks of this approach is that it does not take into consideration negated terms. We also do not leverage prior knowledge regarding prominent terms. The step by step process is shown in Figure 12.

b. Dealing with sparsity

The semantic descriptor is sparse, mainly because the respective annotations contain a small amount of words, as compared to the total number of lead roots. This may affect accuracy and consecutively fail to be a proper representative of the semantic data. A simple trick is devised to make it denser. After the semantic descriptor is established, the probabilities of occurrence of the lead roots in every class are calculated. Some of these words are shown in Table 3. Words that have a probability higher than a certain threshold (in our case we chose 5% heuristically) in at least one of both classes are kept, while the others are discarded. Increasing the threshold leads to less lead roots with more frequency, rendering the semantic descriptor denser. By increasing the threshold, some subtle words that may be indicative of the corresponding class might be lost. This is evident by the lower *Accuracy* obtained with higher thresholds. Table 4 includes the length of the semantic descriptors at different thresholds for the first 10 semantic levels.

Table 3 Probability Table of Sample Terms per Class

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

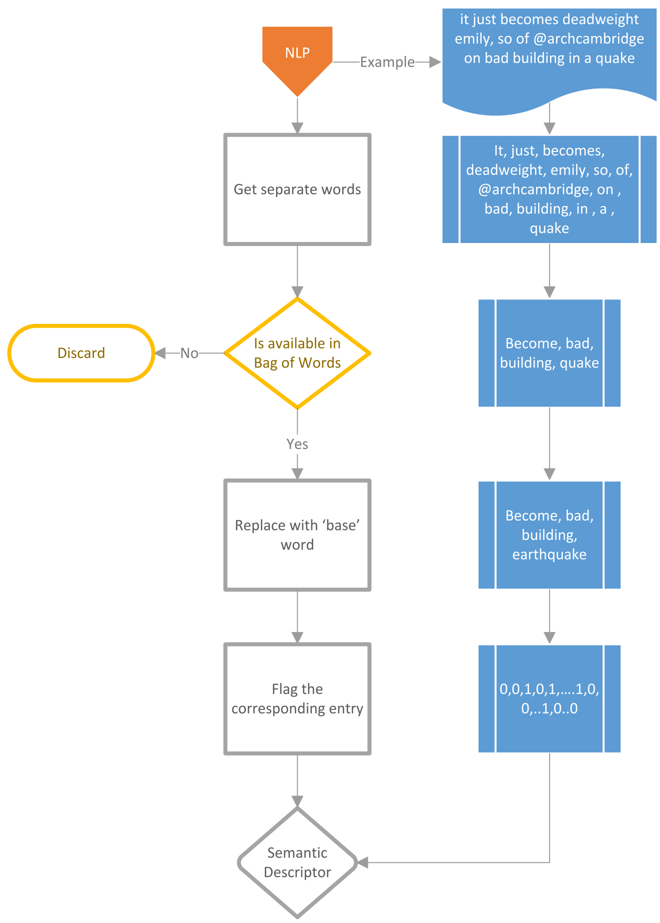


Figure 12 Obtaining the Semantic Descriptor

C Learning

1. Score and Decision Fusion

This proposed learning approach depends on the output obtained from low-level and semantic classifiers independently.

a. Semantic and Low-level Classification, Step 1

Three types of classifiers are investigated for this step, specifically Ensemble Learning, Neural Networks and Support Vector Machines (SVM). Two types of kernels are investigated for SVMs as well, linear (regular), and a Gaussian kernel. Images of damage and their associated annotations are linearly inseparable according to the representations we used. By projecting the features into the Gaussian space, a better accuracy may be achieved. The performance of these classifiers on the proposed visual and semantic features is addressed in the Experiments chapter.

b. Score Fusion and Decision Fusion, Step 2

Augmenting visual and semantic descriptors boosts performance. Stemming from this statement, a hybrid approach is explored. More technically, two methods are proposed; one which relies on the decision fusion of the independent classifiers (visual and semantic), i.e. *Majority Vote*, from *Step I*, and the other method on score fusion of the outcomes, *Score Learning*. It is worth noting that using MATLAB R2015b for classification, the output is expressed in two values, the label, which indicates the predicated class **decision**, and the class likelihood measure, **score**. Throughout the experiments, we approach visual feature combinations in two manners, either the features are aggregated as one vector and trained on one classifier resulting in one outcome, *Two-score Fusion* for learning and *Two-decision Fusion* for *Majority Vote*, Figure 13, or every feature is trained on a separate classifier, resulting in several outcomes, grouped together *Multiple-score fusion* for *Score Learning* and *Multiple-decision fusion* for *Majority Vote,* Figure 14. For example, if we want to explore *Grad* and *Gabor* features with semantic descriptors, the *Two-score fusion* entails that we would use the score from the *Grad-Gabor* classifier resulting in a two-dimensional feature, one from the *Visual Feature* classifier and one from the *Semantic Descriptor* classifier, whereas the second approach, *Multiple score fusion*, would use the scores of the *Grad* classifier and the *Gabor* classifier along with the score from the *Semantic Descriptor* classifier, resulting in a three-dimensional feature, and so on. The decisions and scores utilized at the second step, *Score and Decision Fusion,* are based on the classifiers that achieved the best *Accuracy* from Step *I, Semantic and Visual Classification.*

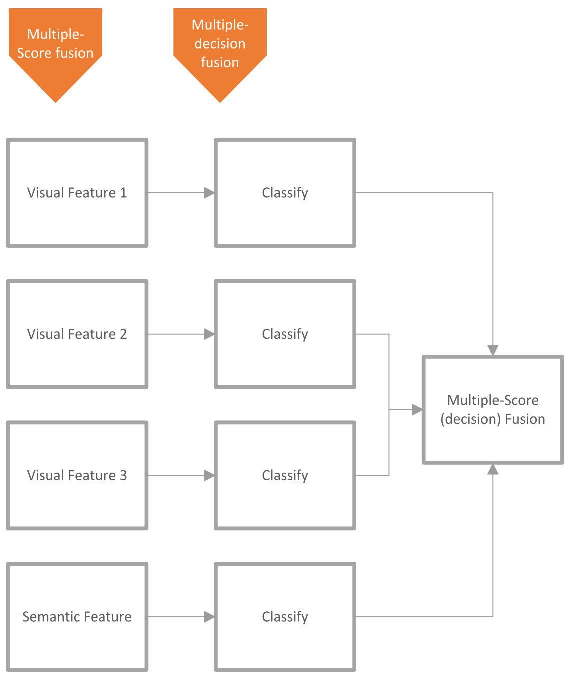


Figure 13 Multiple-Score Fusion and Multiple Decision Fusion

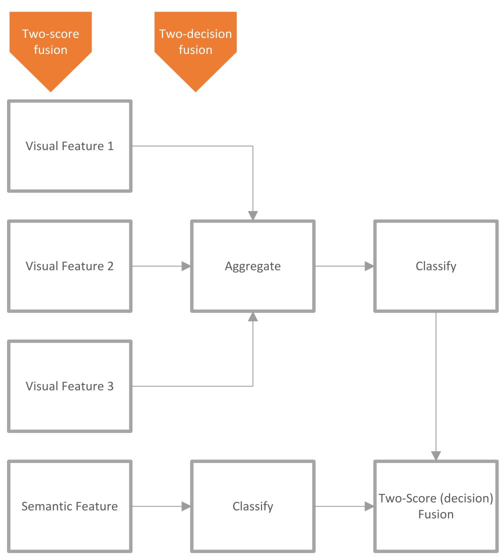


Figure 14 Two-Score Fusion and Two-Decision Fusion

2. Feature Aggregation

Similar to the visual feature aggregation, the semantic descriptors are concatenated with the visual features to form a single feature vector. These features are trained and tested on the same classifiers used previously.

3. Cortical Algorithm Learning

a. Structure

Cortical algorithms (CA) are an artificial computational model, neural network, that mimics the activity of the brain. The architecture of the CA constitutes six hyper-column layers of different thickness neural networks (Edelman and Mountcastle), which is inspired by the cortex of the human brain. The basic structure of this cortical network is the hyper-column, a group of neurons associated to the same sensory input region. Three types of connections govern the cortical network, horizontal and vertical, and between non-consecutive layers. The horizontal connection represents that between the hyper-columns within the same layer, whereas the vertical connection joins the hyper-columns of consecutive layers. CA networks are hence learnt by tuning the weights of the connections. Non-consecutive layer connections are ignored for simplicity. CA are designed and used for supervised classification, however, (Hajj, Rizk and Awad 327-334) employed CA for unsupervised training by focusing the output of the network as a new representation of the observations studied. Training the cortical algorithm for either supervised or unsupervised learning is divided into the following steps.

b. Supervised and Unsupervised CA schemes:

The connections which connect the neurons and layers in the network, referred to as weights are initialized as random values, typically close to 0. The second step is an unsupervised feedforward approach. Feedforward learning is done so that hyper-columns may be able to identify features amongst observations that pass undetected by other approaches. Similar to the action in the human-brain, this is done by inhibiting and strengthening the weights, tuning, based on the input pattern. The weights of a particular column are strengthened whenever the firing pattern coincides with a particular input pattern. The weights of neighboring columns are consecutively inhibited. In this iterative process, the columns learn to fire a specific pattern based on the input, which in turn allows for the extraction of features from the data. This feedforward learning does not ensure the firing of specific invariant patterns for recurring input. This is ensured with the supervised feedback step. When a firing pattern deviates from the specified pattern, a feedback signal is generated, so that the firing column that lead to the misclassification is inhibited., and the column from the original pattern is fired. Ultimately a “stable activation” is reached once the firing scheme no longer changes with multiple exposures. Once this is reach, each layer is trained until convergence.

As mentioned earlier, (Hajj, Rizk and Awad 327-334) propose a distributed implementation of CA on a MapReduce framework for unsupervised clustering. Their proposed approach follows the same initialization and feedforward steps mentioned earlier, however they take the output from the feedforward learning as a new representation of the observation, and assign it to a cluster randomly. Subsequent data points are then assigned to the clusters with the closest centroid, if the distance between the given point and the nearest centroid is below a certain threshold, otherwise the instance is assigned to a new cluster. This step results in scattered clusters that may not be meaningful without fine-tuning and processing. The dispersed clusters that fall within a distance of a predefined threshold, intra-cluster distance, are grouped together. Afterwards the points of each cluster examined. If the farthest points within the same cluster are larger than a threshold, inter-class distances, the cluster is divided into two smaller clusters, and the other points assigned to the closest center. The proposed approach is an iterative process, where it ends when steps 2 and 3 result in stable clusters.

c. Strong CA

Strong CA relies on modifying the elementary computational unit as well the connectivity of the network to include recurrent connections. Unlike traditional artificial neural network where excitation and inhibition alter the weight of a connection based on the response of the cell an input, we adopt a more biologically plausible approach by encoding these processes in the synapses mimicking the role of GABA, the chief inhibitory compound in the mature vertebrate central nervous system, and NMDA receptors, which control the memory function and allow the passage of information once activated. In recurrent connections, the post-synaptic neuron is looped back to the pre-synaptic neuron, so in other terms if a pre-synaptic neuron inhibits (excites) a post-synaptic neuron, the post-synaptic neuron will in turn inhibit (excite) the pre-synaptic neuron. This loop leads to more complex patterns.

A considerable amount of models has been proposed in the literature to model the activity of a single neuron. These models generally fall under two types: biological models which attempt to replicate the firing patterns observed in single cell recordings and artificial models which abstract from the biology and model neurons as a computing unit implementing a simple function. While the former expresses firing rate as an often complex function of an input current, the latter deals with a numerical output in response to a numerical input. In Strong CA, we borrow the concept of strength encoding from the anatomy: the strength of a stimuli is reflected at the output of the neuron by the firing rate observed. In other terms, a "strong input" leads to a higher frequency of spikes at the output of the neuron.

Mathematically, let $[x_1, x_2, \ldots, x_n]_t$ be a vector representing n synapses connecting to a neuron at time t, weighted by the vector $[w_1, w_2, \ldots, w_n]$, $w_i > 0$ representing the corresponding synaptic strength, the output y of the neuron is expressed as:

$$y = \begin{cases} \dfrac{1}{1 + \exp(\sum_i p_i x_i w_i)} \ for\ t = t, t+1, \ldots t+m\ if \sum_i p_i x_i w_i > \theta \\ 0\ for\ t = t\ else \end{cases}$$

The model states that the response of a neuron to the total stimulus is a train of spike sustained for m time steps where $m = \tau\ \sum_i p_i\, x_i\, w_i$. Following this train of spikes is a refractory period, represented by r time steps of zero activity, where no input can excite the neuron to generate an action potential. $p_i = 1$ if the synapse is excitatory and 0.01 if the synapse is inhibitory. In other terms, during the refractory period, the neuron is unresponsive to stimuli.

Additionally, we introduce recurrent connections linking the output of the neuron to its input through a weighted connection which weight (as well as all weights in the network) will be learned according to the unsupervised scheme proposed in (Hajj, Rizk and Awad 327-334)

CHAPTER IV

DATABASE

This section details the steps that led to creating the dataset: crowdsourcing through Twitter, building the database, and labeling the classes, as well as some data set statistics and examples.

A Crowdsourcing through Twitter

Twitter is one of the most widely accessed social media networks, which is used as a platform to 'tweet' messages and 'retweet' them, with an option to tag a media element such as a small clip or an image. Twitter ("Twitter.")has approximately 310 million monthly active users with 83% of them on mobile devices. This is a considerable amount, which makes almost every corner in the world reachable. It also gives access to new events in a matter of a click of a button, which during these times is faster than the conventional methods represented by the news channel or the newspaper. Having such accessible big data can be useful on many levels that require fast and accurate response.

B Building the data set

Building the dataset was done solely based on twitter feed. The process was done by querying keywords such as: earthquake, damage, disaster, crisis, flood, etc., on a Java-developed tool that uses Twitter API to access Twitter. The tool collects the tweet in the form of a JSON object, and parses the associated message. Attachments are tweeted in the form of URL's. This media URL which contains the image is then downloaded via MATLAB, and saved along with the respective description.

C Labeling

Typically, the gathered images contain noisy elements, consisting of random pictures that convey no damage-related visual information. This can happen mainly because tweets are retrieved through keywords, and not based on visual content of the associated media file. The images that don't qualify as damage images are discarded manually, as well as redundant images that pass in the form of retweets. After removing images do not convey any concept of damage, for example, people taking selfies, clean natural environment, and urban places, we identified the classes. It should be noted that the term 'damage' is a fuzzy word that has no specific visual representation. For example, fallen tree can be coined damage, just as sincerely as broken wall. Nevertheless, there seems to be an inter-annotator agreement, where people can differentiate types of damage easily when facing them. In this sense, a limited crowdsourcing was used to provide some textual info. Three participants (age around 25 majoring in Electrical Engineering pursuing graduate degrees in the Machine Learning track) were asked to independently separate the images in the dataset based solely on the visual content. The resulting sets of images were compared to each other and to their respective images by a fourth participant.

The images are classified by this group into 2 major classes: 'built-infrastructure damage' and 'nature damage'. Built-infrastructure damage includes the images representing damage to man-made objects, i.e. houses, buildings, walls, etc., whereas nature damage represents, as the name suggests, damage that hit natural objects, i.e. trees, land, crops, etc. Unlike other designed datasets, the images in our homegrown set represent a diverse variation of the label concepts. This is one of the limitations of crowdsourcing through social media, where the gathered observations do not particularly share similar conditions in terms of camera angle, focus, scope of the scene, noise within the image itself and other characteristics.

D Database Statistics

Some examples of images that belong to the identified classes are displayed in Figures 14 and 15, followed by a sample of the annotations for each class. Finally, this section also contains Table 5 highlights some information regarding the resulting dataset. The database was collected from tweets sent out between February and May 2016, around which earthquakes hit Nepal, Chile, and Japan, and floods hit Kenya.

Table 5 Database Information

Class 1 Infrastructure Damage	1077
Class 2 Nature Damage	271
Imbalance Ratio	3.975
Low-Level Attribute Characteristics	Integer, Gaussian Distribution
NLP Attribute Characteristics	Boolean
Associated Tasks	Classification

government, Nepal, quake, survivors, hits, warning

Nepal, give, rebuild Quakehit airport

Cost, damage, lake examines, spotted

Biggest, major, earthquake

Nepal, quake, areas, deliver, stronger

City, house, learn earthquake

Deadweight, bad, building, quake

Quake, disaster, triggered, changes

Figure 15 Samples of Built-Infrastructure Damage Images and Phrases

storm, damage, area, winds

Wind, damage, reported gust

Hurt, tornado, winds, damage

Nature, creates, important

Infrastructure, damage, assess

Storm, damage, county, severe

Swim, lava, river

Storm, damage

Figure 16 Samples of Nature Damage Images and Phrases

CHAPTER V

EXPERIMENTS

In this section, the experimental setup and results are presented. To test the accuracy of various classifiers, a 5-fold cross-validation is performed globally, i.e. all classifiers are trained and tested on the same data. To measure the performance of the classifiers two metrics are used, the normal accuracy measure to evaluate the accuracy of the classifier for each class, and the F-measure to compare the overall performance of the classifier. In other words, the F-measure is used to evaluate the trade-off between improving the accuracy of the negative class and accuracy loss of the majority class. Here the classifier with higher F-measure is considered to be more accurate. The database crowdsourcing authenticity is compared with a modified cortical network developed for clustering, *Strong CA*, as mentioned in the Methodology Section. In this work, the negative class samples are considered the 'Nature damage' sample whereas the 'Built-Infrastructure damage' samples are considered the positive samples. Misclassifications are also addressed latter in this section. The simulations are executed MATLAB R2015b on a computer with Intel(R) Core(TM) i7-4700MQ and 2.4GHz CPU and Windows 10 64-bit operating system.

A Standardizing

The extracted visual features consist of real values that vary in magnitude. This fact, if left unnoticed, can have severe impact on the training process, mainly due to the biasing that occurs when learning the weights. This point is easily solved through standardizing. For every training fold, each feature is normalized into a Gaussian distribution with mean value of zeros and standard deviation of 1. The testing fold is then projected to the respective distribution. This step ensures that the predictors are insensitive to scales on which they are measured.

B Damage vs Non-Damage

Before going into damage 'type' classification, we would like to make sure that damage, as a concept in itself, can be automatically identified. To do so, a set of images unrelated to damage from the original hand labelling, labelled as non-damage images, are used for training and testing. These non-damage images were collected by querying the same keywords originally used to gather damage images.

1. Classification based on Visual Features

Figure 17 highlights the best accuracy of visual feature combinations that are used to differentiate between damage and non-damage images. It is safe to say that the classifiers are able to easily differentiate between these two classes, considering the relatively high accuracy for most of the feature combinations. The best amongst all was the Gaussian kernel SVM classifier that achieved *Precision* of *92.22%*, an *F-measure* of *93.25%*, and an *Accuracy* of *93.01%* by using Gabor-GIST-Grad-HSV-RGB feature combinations.

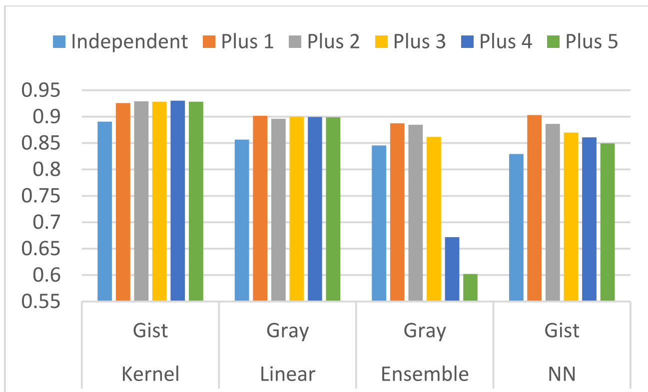


Figure 17 Accuracy of visual feature combinations on Damage vs Non-Damage

2. Classification based on Semantic Descriptors

In Figure 18, the *Accuracy* of the best semantic descriptors is shown across classifiers. Similar to visual feature classification, using a Gaussian kernel SVM led to the best *Accuracy* by projecting the annotations on the 1st bag of words, i.e. *Data01*.

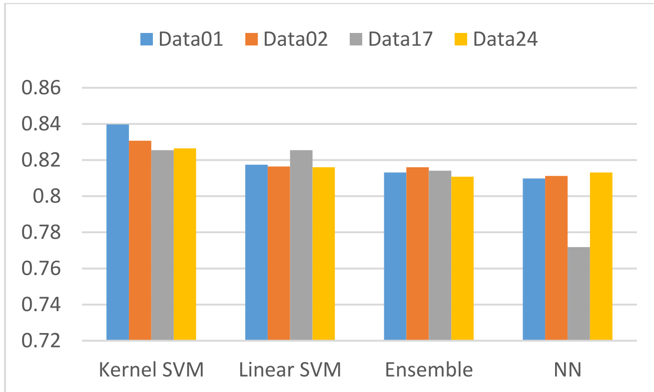


Figure 18 Damage vs Non-Damage semantic descriptors accuracy

C Semantic Classification

1. Proposed Semantic Descriptor

The semantic descriptors are established after several pre-processing steps, which involves creating bag of words from synsets gathered from WordNet, filtering of these bag of words, and using an empirical threshold of *5%* to reduce the amount of observations (lead roots). As mentioned in the Methodology, 50 semantic descriptors corresponding to 50 semantic levels are created. However, for the sake of visualization, only the first 10 results are presented. The semantic features are named *Data01, Data02,* etc., where the last two digits represent the semantic level of the descriptor. Similar to the visual features, the semantic descriptors are trained and tested with 4 distinct classifiers. The *F-measure* of these descriptors is highlighted in Figure 19 and addressed in the following sections, nonetheless other statistics such as *Precision* and *Accuracy* are also reported.

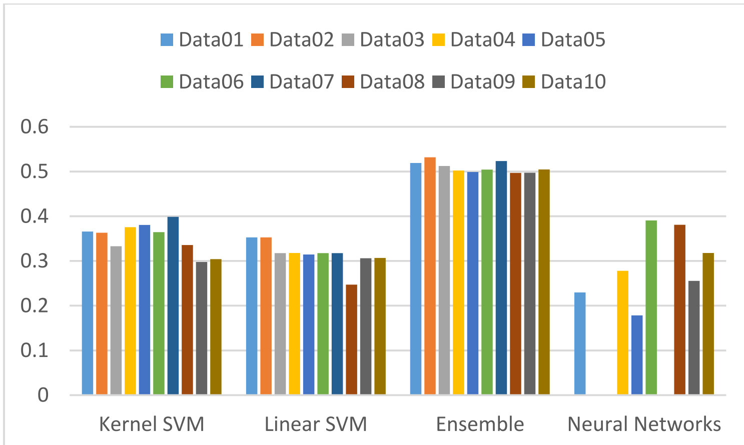


Figure 19 F-measure of Semantic Descriptors

a. <u>Kernel SVM</u>

The precision using Kernel SVM increases between the 1st and 2nd semantic levels followed by a significant drop at the 3rd semantic level (*11.48%*), where it again increases and decreases to the lowest accuracy. The best performance is achieved with *Data07*. With an *Accuracy* of *83.53%*, it is not that higher than the worst *Accuracy*, *80.93%* for *Data09*, nonetheless 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.

b. <u>Linear SVM</u>

The *Accuracy* using this classifier is almost consistent with a standard deviation of *0.3%.* The *Sensitivity* is even lower than Gaussian Kernel SVM, which affects positive class classification. This is mainly due to the fact that several phrases, as highlighted in the Error Analysis Section, are in common between both classes. This fact confuses the classifier that fails to learn that one phrase can belong to the both classes.

c. Ensemble Learning

Ensemble learning is much more sensitive to positives than the SVM classifiers. The *Accuracy* obtained using Ensemble learning testing is incredibly low in comparison to the other classifiers. On the other hand, Ensemble Learning resulted in the highest F-measure.

d. Neural Networks

Neural Network architecture used here consists of one hidden layer with number of neurons equal to that of the descriptor dimension. The results shown in Figure 18, are missing *Data02, Data03, and Data07,* since no true positives where rightfully classified during testing. The performance of other descriptors is acceptable, but still less than Gaussian Kernel SVM.

e. Discussion of Semantic Classifiers

Amongst all 4 of the Semantic Descriptor classifiers, the Gaussian Kernel SVM had the best performance, with an *Accuracy* of *83.53%* via *Data07*. Based on these results, the outcome of this particular classifier is used for Score and Decision Fusion.

2. Word2Vec Descriptors

The proposed semantic descriptor scheme mentioned earlier involves intermediary processing steps, passing through a bag of word and applying an empirical threshold to remove noise. In this section we address raw semantic data processing. This is done by utilizing the tool developed by (Mikolov et al.), publically available as word2vec. Word2vec utilizes continuous bag-of-words and skip-gram architectures for computing vector representations of words. A text corpus is used for training by constructing a vocabulary from the training text data and then learns vector representation of these words in terms of numbers. Untrained words can be projected on the output, and represented in turn using numeric values. For our training purposes, we used the first billion character from Wikipedia, available online on Matt Mahoney's page.(Mahoney) The dimensionality of the word vector representation can be user-defined. In our experiments, we used the same classifiers applied on the proposed semantic desriptor. 5 feature dimensions are created, with 1, 2, 5, and 10 numeric values. Since the number of words differs between observations, the semantic representation of each observation was extended to the length of the largest amount of words present in these observations. After inserting the numeric values for each words, the rest of the feature vector is zero padded. The *Accuracy* are presented in Figure 20.

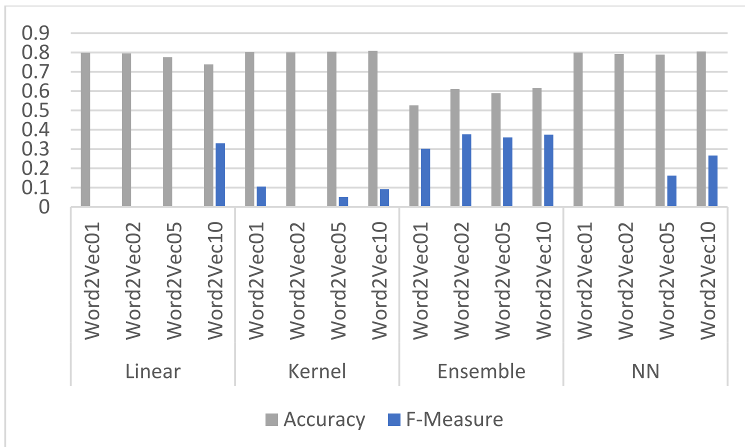


Figure 20 Accuracy of Word2Vec classification

The results appear to favor an increase in the number of values representing each world. This is obvious from the *Accuracy* of *Word2Vec10*, where every word is represented by 10 numeric values. 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 amongst three out of four classifiers used.

D Low-level Classification

Pairing 6 distinct features together in every possible way leads to 63 separate groups. All these variations are trained and tested. The statistics, are the highest for every value of pairing, i.e. *Plus 1* for the *Grad* feature is the highest amongst all other pairs of the *Grad* feature with other features, and so forth. Figure 21 represents the *Accuracy* and *F-measure* of the best group of combinations of visual features per classifier respectively.

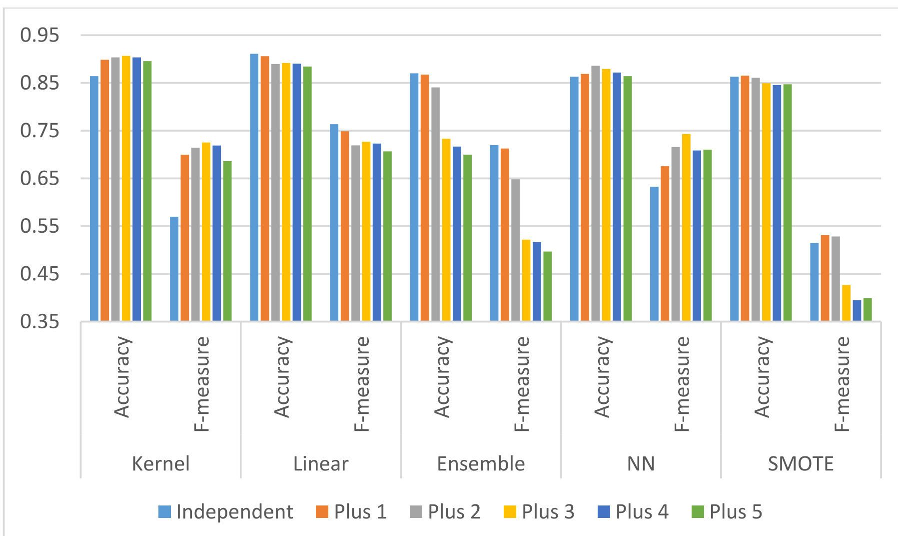


Figure 21 Performance of feature combinations on homegrown dataset

1. Kernel SVM

A general conclusion that can be drawn at a first look upon the charts is that the *Accuracy* increases as a function of concatenated features. The lowest performance is achieved using RGB feature vector independently with an *Accuracy* of *83.46%* and an *F-measure* of *42.20%*. The best performance however is achieved by using *Gabor Plus 3*, where the 3 in this case are *Gist*, *Gray* and *HSV*. The best *Accuracy* is *90.65%* with an *F-measure* of *72.50%*. The relatively inferior performance of the *RGB* feature may be certainly attributed to the fact that the images from both classes share a lot of colors. Granted that some '*Nature Damage'* images, as the name suggests must contain more green, then let's say gray, green can be also present in abundance with *'Infrastructure Damage'* images. The *RGB* feature also affected the performance of other combinations. This is noticed in the decline of performance between all the *Plus 5*'s, except for the *RGB Plus 5* itself, where combining all other features with *RGB* boosted the performance.

2. Linear SVM

The same set of combinations are trained and tested on a regular SVM. The *Accuracy* and *F-measure* follow the same trend. Unlike the Gaussian Kernel SVM, *Gray* features start off with an *Accuracy* of *91.10%*, and then decreases gradually. The.

Gabor rises suddenly around *6.01%* then descends, while other combinations of features follow the same ascending trend.

3. Ensemble Learning

Ensemble learning melds results from many weak learners into one high-quality ensemble predictor. Because the data set has an imbalance ratio of *3.971*, RUSBoost is especially effective with imbalanced data sets. Entries from the 'major' class are under-sampled, and then learning is boosted using AdaBoost. In Figure 21, It appears that Ensemble learning is not as effective as any of the SVM's. Some feature combinations, specifically the *Gist, RGB,* and *Grad* increase incrementally, but the overall *Accuracy* is not encouraging. On the other hand, *Gray, Gabor,* and *HSV* feature combination start off with a relatively high *Accuracy* then drop when augmented with other features.

4. Neural Networks

Neural networks are also explored to find a latent connection between the probabilities. In this attempt, we train several neural networks using backpropagation and the best *Accuracy* is obtained via *Gabor-Gray-HSV.* In choosing the architecture of the Neural Network, one hidden layer is created where the number of neurons is equal to the vector size of the feature. The *Accuracy* of the Neural Network follows the same trend as that of the Gaussian Kernel SVM, i.e. the *Accuracy* enhances as a function of features augmented. This leads to the conclusion that the more feature spaces are used to represent the image, the more hidden connections can be made to increase *Accuracy*.

5. SMOTE SVM

Oversampling the negative class data via SMOTE resulted in *Accuracy* increase as a function of features as well. Nonetheless, the best *Accuracy,* which is the result of aggregating *Gabor* and *Gray* features, was still less than that of all the other classifiers, *86.50%,* shown in Figure 21.

6. Discussion of Visual Classification

Judging by the performance of these classifiers, Kernel SVM is adopted for the Score and Decision Fusion. Although the best *Accuracy* obtained with Linear SVM, *91.10% vs 90.65%* for the best Kernel SVM *Accuracy,* the latter achieved higher *Precision* against the former, *88.78% vs 81.24%.* The F-measure of Linear SVM is *76.34%* vs *72.50%* for the Kernel SVM.

7. Visual Classification on a Different Dataset

In order to demonstrate the effectiveness of these features, they are tested on a separate dataset. The dataset used for comparison is created using the SUN Database (Xiao et al. 3485-3492) for images. The SUN Database is a designed data set of images gathered from several search engines, describing scenes by labels which satisfy the statement: “I am here.”. It encompasses 397 scenes and is also labeled based on the object content. Since the proposed approach handles *Infrastructure damage* and *nature damage*, the closest thing for comparison was to test *City* versus *Landscape*. The *City* images consisted of '*rubble, office building, city and building façade*', whereas the Landscape images consisted of 'archaeological excavation, bog, forest and forest road'. The result is 740 *City* images as opposed to 256 *Landscape* images. No preprocessing was done on the images except resizing them to 256×256 . By taking a look at Figure 22, we notice that the performance of the classifiers on this dataset follows the same trend as the performance on the homegrown database described earlier. Specifically, the *Accuracy* increases as a function of features aggregated, with the *RGB* features increasing the most. The best performance in terms of *Accuracy* is obtained by the *Kernel SVM* (*96.69%*). In terms of *Precision, Kernel SVM* also dominated with a *96.25% vs 93.17%* achieved through *Linear SVM.* Unlike the homegrown dataset, *SMOTE SVM* performed better than Neural Networks in terms of *Accuracy*, with *94.38%* vs *90.46%.*

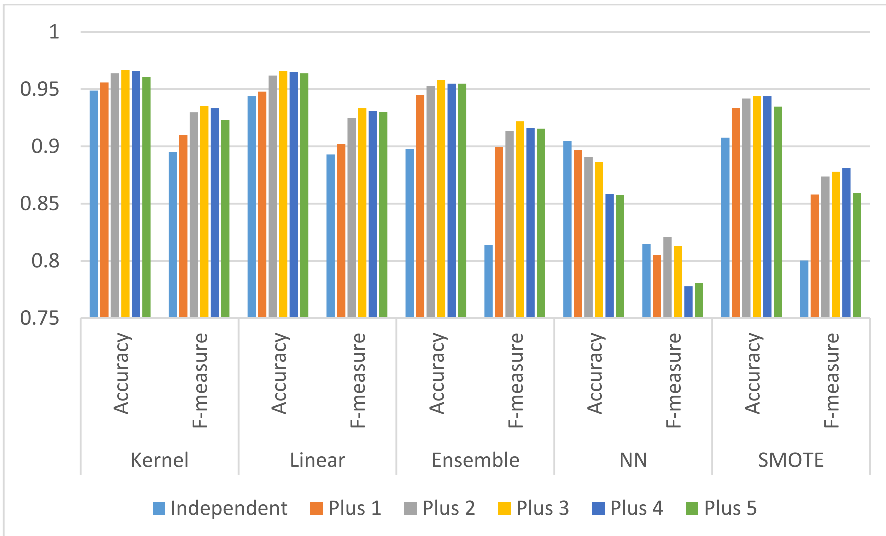


Figure 22 Performance of feature combinations on SUN database

E Score Fusion and Decision Fusion

The output of the *Step 1 classifiers* is expressed in two values, the label, which indicates the predicated class **decision**, and the class likelihood measure, **score**. Two approaches are sought at this stage, the first utilizes the decision of the *Visual Features'* and *Semantic* classifiers through majority voting, while the second aggregates the scores of these classifiers as new feature vectors to be trained, *Score Learning.*

1. Majority Vote

In this approach, the predictions of the visual feature SVM and that of the semantic descriptor SVM are utilized. These predictions are combined, through an AND logical function, where the prediction is positive if and only if both SVM's agree on it. This step is straightforward, yet still improved the *Accuracy* as shown in Figure 23.

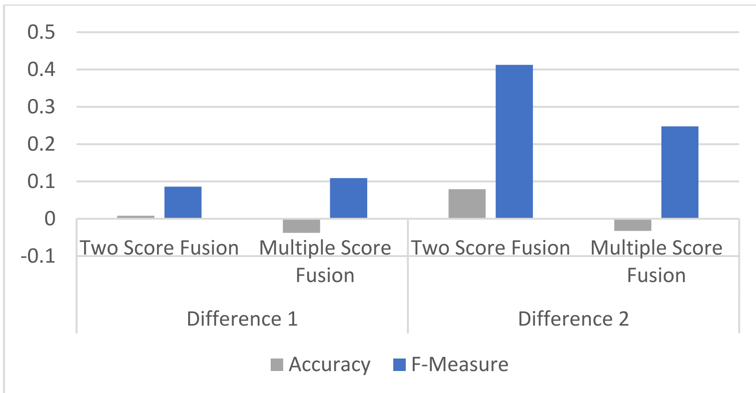


Figure 23 Difference in Accuracy between Augmented feature and Corresponding visual feature

Figure 23 shows the differences between the best *Accuracy* of augmented features, through *two score decision* and *multiple score decision*, and the corresponding visual feature (alone), *Difference 1*, and the semantic descriptor which was augmented, *Difference 2*. *Two score decision's* best *Accuracy* was achieved by *Gabor-Gist-Gray-HSV-Data07*. Evidently, the *Accuracy* obtained using *two-score fusion* increased, albeit a mere *0.081%*, while the *Accuracy* using *multiple-score fusion* decreased. Regardless, this hybrid approach favors *Sensitivity* and *F-measure*, while sacrificing *Precision* and *Specificity*. This can be expected since the AND function will affect the TP rate, whenever the outcome of both SVM's is different. On the other hand, by using the *Multiple score decision* scheme, it turns out that sticking to a single *Visual Feature*, specifically *Gabor,* with *Data01* achieves the best *Accuracy*. Nevertheless, the overall performance doesn't improve the *Accuracy* as compared to the *Visual Features* and *Semantic Descriptors*, shown also in Figure 17. The *F-measure* however is around *10.09%* better against *Visual Features* and *24.76%* better against *Semantic Descriptors*.

2. Score Learning

The second approach is more about training new classifiers based on the scores obtained from the previous SVM's. In a sense, the independently trained SVM's act as a kernel, that reduces the features to scores, resulting in a small dimensional feature vector, one score from every classifier. Generally, the performance of this newly suggested SVM relies heavily on the performance of the preceding two. If the *Gaussian Kernel SVM* that lead to the probabilities utilized did not have a high *Accuracy* rate in the first place, it would be futile to even suggest such an approach. Both *Two-score fusion* and *Multiple score fusion* methods are used here. Figure 24 combines the statistics of the best results achieved using different classifiers with *Two-score fusion*, whereas Figure 25 presents the difference between *Two-score fusion* and *Multiple Score Fusion*, *D1*, and *Two-score fusion* and *Feature Aggregation, D2.*

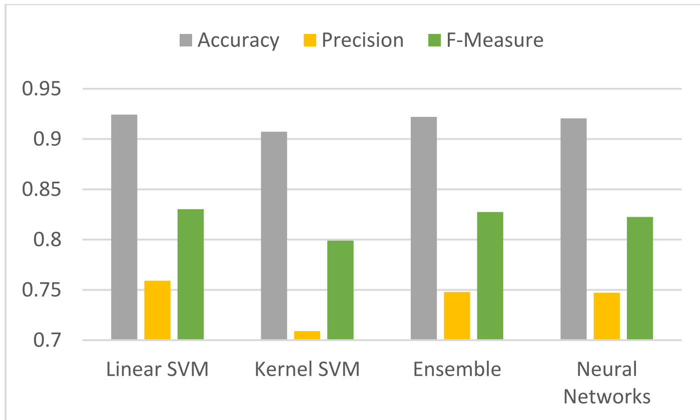


Figure 24 Performance of Two-Score Fusion

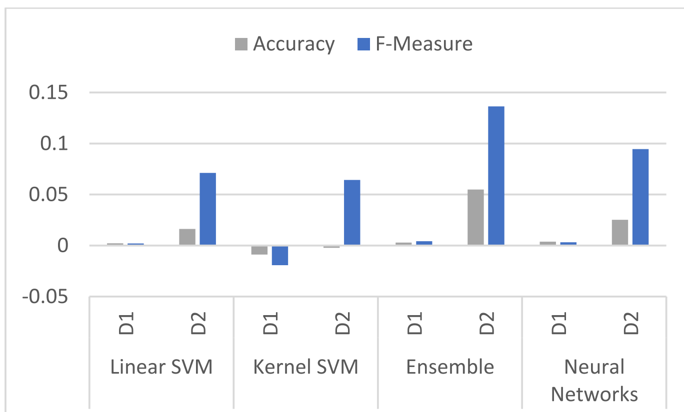


Figure 25 Difference D1 in Accuracy between Two-Score and Multiple-Score Fusion, and D2 between Two-Score Fusion and Feature Aggregation

Figure 26 highlights the difference between the different approaches and their corresponding visual features.

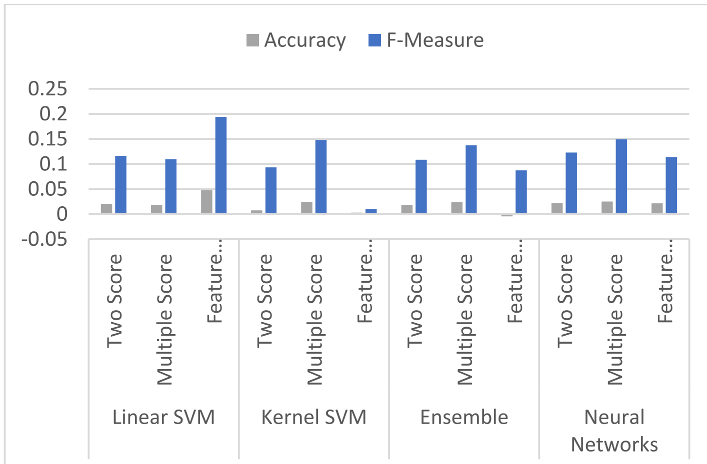


Figure 26 Difference in Accuracy between different classifiers and their corresponding visual features

It is clear that augmenting the semantic features to any visual combination, improves the performance. The individual classifiers are discussed below.

a. <u>SVM</u>

A Gaussian Kernel SVM and a regular SVM are both trained on the probabilities, and their performance is compared to the corresponding visual SVM performance. The best augmented performance was achieved using different set of independent features. The best *two-score fusion* accuracy via linear SVM is obtained using *Gist-Gray-HSV* visual features and *Data01*, as well as with *Gabor-Gist-Gray-HSV* and *Data01*, but since the former contains less features, we emphasize it instead. On the other hand, *Gist-Grad-Gray-HSV* visual features combined with *Data03* gave the highest kernel SVM *Accuracy*. Training the SVM on the probabilities reduces the

Precision and *Specificity*, but has higher *Accuracy* and an even better *F-measure*. The linear SVM shows better *Accuracy* than the Gaussian SVM in terms of *Accuracy* and *F-measure*. This may be attributed to the values of probabilities, which should in the first place be linearly separable, i.e. below zero's should belong to negative class and above zero's should be positive class. By using the *Multiple score fusion*, the best Linear SVM Accuracy is achieved by using the scores of *Gabor-Gist-Grad-Gray-HSV* independently obtained from the visual features' classifiers along with the *Data04* score. The *Accuracy* and *F-measure* are almost, yet less, as those of the Linear SVM of the aggregated features, *92.21%* and *82.82%* vs *92.42%* and *83.02%* respectively. Unlike the Linear SVM comparison, the Kernel SVM applied using *Multiple score fusion* is better than kernel SVM using *Two-score fusion*. The best *Accuracy* is achieved by *Gist-Grad-Gray-HSV-RGB* along with *Data06.* It took combining all the features to achieve this result.

b. Ensemble Learning

Ensemble learning is obtained by fitting an ensemble via RUSBoost and 20 weak learners. The best *two-score fusion Accuracy* is achieved with *Gabor-Gist-Grad-Gray-HSV* visual features with *Data01.* It turns out that using Ensemble learning, in this case RUSBoost, does indeed increase *Accuracy* and makes the classifier more sensitive to positives. Nonetheless, it affects the negative class classification with a reduced

Specificity value, and lower *Precision*. The best Ensemble classifier used with *Multiple score fusion* is obtained by using the outcome of all the visual features along with *Data06*, just like what happened with Kernel SVM trained through *Multiple score fusion*.

c. Neural Networks

Achieved for *two-score fusion* by using *Gabor-Gist-Grad-Gray* visual features with *Data02*, with its corresponding visual SVM *Accuracy*. The neural network architecture used is a single hidden layer with number of neurons equal to the size of the feature dimension. Again, the *Specificity* and *Precision* are reduced, whereas both *Accuracy* and *F-measure* are still better.

Using the *Multiple score fusion* with Neural Networks faired poorer than the former in terms of *Accuracy* and *F-measure*, and was achieved by using 5 visual features, namely *Gist-Grad-Gray-HSV-RGB* with *Data05*.

d. Two-score vs Multiple-score Fusion

Judging by the *Accuracy* and *F-measure*, the best *Accuracy* for both *two-score fusion* and *multiple-score fusion* was achieved using a linear SVM to train on the probabilities. As mentioned earlier, this might seem straightforward since the new features (probabilities), are linearly separable.

F Feature Aggregation

In this section, the results of feature aggregation are discussed. Figure 24 includes the difference in statistics between the *two-score fusion* classifiers and the classifiers trained on aggregated features. Figure 26 shows the difference between the aggregated feature classification results and the corresponding visual features. Aggregating *Visual Feature* and *Semantic Descriptors* increases *Accuracy* of the SVMs, with Linear SVM showing better differential results than Gaussian Kernel SVM. Ensemble learning achieved the best *Accuracy*. Nevertheless, in terms of *Accuracy* the performance decreased slightly while still maintaining a higher *F-measure*. Performance is also boosted with neural networks, which resulted in a *11.39%* increase in *F-measure,* but still lost *2.7% Precision*.

G Discussion of Supervised Training Results

In this subsection, we take a look at the best set of features for every classifier, Table 6. The table shows one thing in common amongst the results, *Gray* visual feature being part of every combination. This feature conveys statistical information about the texture of the image, regardless of its color. Semantic descriptors on the other hand did not seem to have any significance in terms or re-occurrence in the best features. However, we note that *Data06* and *Data01* both occurred as the best semantic features three times and *Data03* twice with Kernel SVM. Another remark is the fact that for Linear SVM, Ensemble Learning and Neural Networks, feature aggregation (F. Aggregation) had the lowest *Accuracy* where on the other hand using Two-score fusion (Two-score fusion) had the highest.

Table 6 Best Classifier Performances Summary

Kernel SVM	Visual	Semantic	(%)
Two-score fusion	Gist-Grad-Gray-HSV	Data03	9072
Multiple-score fusion	Gist-Grad-Gray-HSV-RGB	Data06	9162
F. Aggregation	Gabor-Gist-Gray-HSV	Data03.	9095
Linear SVM	Visual	Semantic	%
Two-score fusion	Gist-Gray-HSV	Data01	9243
Multiple-score fusion	Gabor-Gist-Grad-Gray-HSV	Data04	9221
F. Aggregation	Gabor-Gray	Data09	9080
Ensemble Learning	Visual	Semantic	%
Two-score fusion	Gabor-Gist-Grad-Gray-HSV	Data01	9221
Multiple-score fusion	Gabor-Gist-Grad-Gray-HSV-RGB	Data06	9192
F. Aggregation	Gray-HSV	Data06	8672
Neural Networks	Visual	Semantic	%
Two-score fusion	Gabor-Gist-Grad-Gray	Data02	92.06
Multiple-score fusion	Gist-Grad-Gray-HSV-RGB	Data05	91.69
F. Aggregation	Gabor-Gray-HSV	Data01	89.54

H CA Clustering

Unsupervised learning techniques can be useful when attempting to identify coherent set of observations without prior information about their actual label. In this section, damage types are being clustered based on a model that mimics the network in the human brain.

1. Visual Clustering

As human beings, we see the image as raw pixels, which is exactly what we attempt to cluster. However, for comparison purposes, the visual features that achieved the highest supervised classification results, *Gabor-Gist-Gray-HSV*, also undergo clustering. The resulting two clusters are assigned to one of the classes of the damage type based on the label of the observations that are linked together. The cluster that includes the most type of observations with the same label, is used as a reference to said label. We investigate the regular CA approach, the Strong CA, and K-Means clustering algorithm to serve as a benchmark for the results.

In Figure 27, we notice that the regular CA performs rather well in comparison with the conventional K-means in terms of *Accuracy* and *F-measure*. However as predicted, by remodeling the neuron function and tuning the parameters of the CA network itself, *STRONG CA* demonstrates better performance, with *12.91%* rise in *Accuracy* and *18.82%* increase in *F-measure* on raw pixel data.

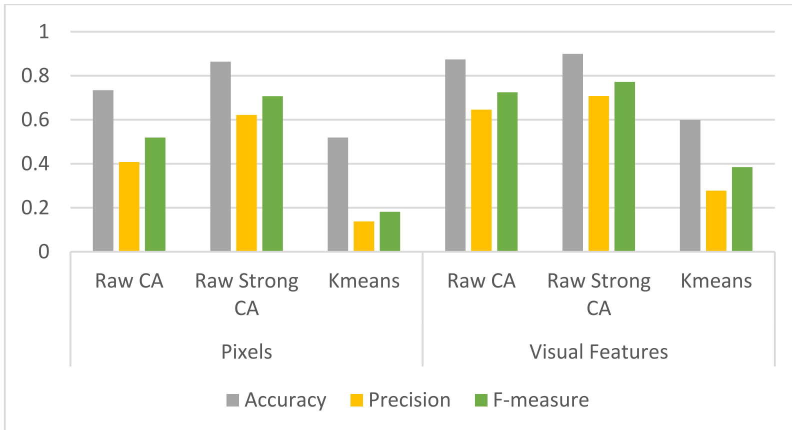


Figure 27 Performance of Clustering algorithms on pixel data and on visual features

A good understanding of the *CA* approach requires us to implement it on the visual features as well. In Figure 27 also, the *Accuracy* of visual feature clustering is also presented. Clearly from the graph, we can deduce that the visual features extracted result in better clusters. We can witness quasi-consistent increase in *F-measure* of around *20%*. Although regular CA *Accuracy* on visual features shows a slightly higher increase that using *STRONG CA*, *STRONG CA* still compares better with an *Accuracy* of *89.91%* vs *86.35%* for *CA*.

2. Semantic Clustering

We also attempt to cluster the annotations associated with the images. Here also we focus on the semantic descriptor the achieved the best *Accuracy* in semantic classification, *Data07*. Figure 28 includes the *Accuracy* of *Data07* on the different clustering approaches.

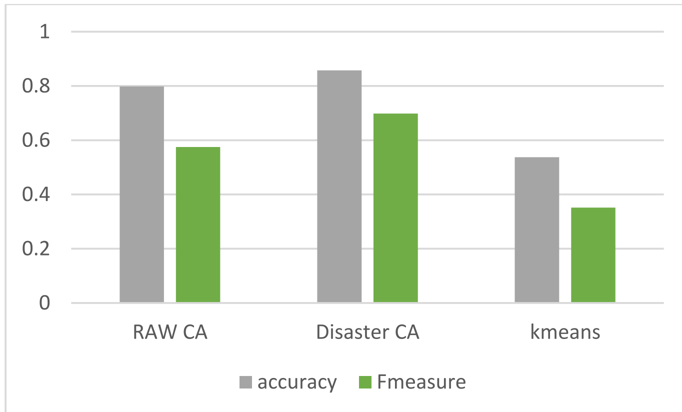


Figure 28 Performance of clustering algorithms on semantic data

Similar to the visual clustering, *STRONG CA* ranks chief in comparison with the regular *CA* method, while K-means shows a significantly lower *Accuracy*. The *Accuracy* achieved here, *85.76%*, is also higher than that obtained from supervised classification, *83.53%*, and a greater F-measure.

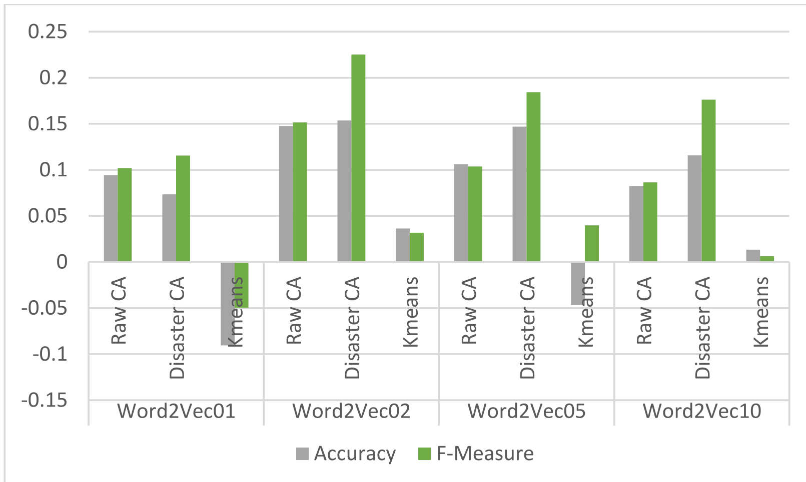


Figure 29 Difference in Accuracy between proposed semantic descriptor and Word2Vec descriptors

Figure 29 represents the difference in *Accuracy* between the proposed semantic approach, and the Word2Vec representations. The positive difference signifies that clustering the semantic descriptors yielded better results than any of the Word2Vec representations. Notably, the largest difference is always at *STRONG CA*, which might be attributed to the fact the proposed semantic descriptor *Accuracy* improved the most when clustered with *STRONG CA*. It is also noteworthy that the as the number of numeric value representation of the words increases from 2 to 25, the *Accuracy* on the regular *CA* and *STRONG CA* improves. Opposite to supervised classification, representing a word with one numeric value, *Word2Vec01* lead to the highest *Accuracy* among other representations. This can be explained by the inter- and intra-class distances that are scalable.

I Error Analysis

In this section, we take a closer look at the misclassifications, visual, semantic and augmented, that correspond to the classifiers with the best performance in terms of *Accuracy* and *F-measure.*

1. Visual Kernel Classification

Attached in the table are some misclassified entries. These misclassifications have been grouped, as best as possible, into visually consistent groups, in an attempt to identify the cause behind misclassifying these subcategories. Some of these errors are shown.

Table 7 False Positives of Best Visual Classifier

A

B

C

D

Table 7 contains some examples of false positives, images that were supposed to be classified '*Nature Damage'*, but didn't. Generally, there is some type of consistency amongst the errors. In part A, all the images share an open field view with garbage (visually scattered colors); B shows houses in an urban location with slight damage (broken trees); C is also a group of broken branches, but this time in nature scenery as opposed to the countryside; Finally, in D, groups of people are present in the damaged locations. False negatives are not as high as false positives; *21 vs 105.* Nevertheless, a pattern may be found as well.

Table 8 False Negatives of Best Visual Classifier

A

B

In Table 8, part A seems to include infrastructure damage to property in a nature environment. The abundance and continuity of the green color is evident; part B share a 'flat' ground in nature environment. The absence of chunks of rubble or broken buildings may have caused the misclassification.

2. Semantic Kernel Classification

This subsection tackles the semantic misclassifications. In Table 9, the misclassified phrases are highlighted. TP and TN represent the frequency of each phrase in the 2 classes, whereas FP and FN are the number of phrases that have been misclassified, and R is the ratio of TP over TN. Take for example the phrase 'bring, damage, storm', it is associated with 5 '*Nature Damage*' images and only two '*Built-Infrastructure Damage*' images; i.e. ratio of *0.4*. When training on the sematic data, this phrase is biased towards the negative class, and subsequently misclassified.

Table 9 Sample Semantic Misclassification phrases

Phrase	TP	TN	FP	FN	R
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
damage, quake, toll	2	2	2	2	1

Some phrases that have occurred at least once in the negative class and the positive class are misclassified as positive, for example *kill, earthquake, people, least.* The phrase '*damage, storm, reported*' is present twice as much in the negative class than the positive class, *ratio* of *0.5*. During testing, the positive phrase is misclassified, and the one of the negative phrases is also misclassified. These examples highlight the fact when it comes to describing damage images on Twitter, 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 only one class, which is not enough for learning.

3. Score Learning Classification

The best augmented results are obtained through applying a linear SVM with *two-score fusion* on the scores of the entries, with an *Accuracy* of *92.43%*. Some misclassified images and their corresponding annotations are shown in Table 10. It is worth mentioning however that performance of Two-Score Learning using Linear SVM for *Gabor-Gist-Gray-HSV-Data01* resulted in the same performance of that of *Gray-Gist-HSV-Data01,* however since the latter performance is achieved with less feature representation, we emphasize it instead.

Table 10 False Negatives of Best Augmented Classifier

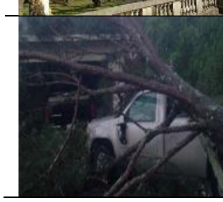
Group 1 Images	Group 1 texts	Group 2 Images	Group 2 texts
	Rise, damage, earthquake		Rise, death, earthquake, year, toll
	death, damage, earthquake, More, dead		Kill, lay waste to, damage, earthquake, seashore, magnitude, people

	rise, death, earthquake, seashore, strike, toll, powerful		Lay waste to, earthquake
	rise, death, damage, earthquake, survivor, toll		Rise, country, damage, earthquake, hit, more, magnitude, toll, people, dead

Just as for the visual misclassifications, some of the images in augmented classification share visually similar traits, however this time, they share semantic words as well. The column of images to the left, Group 1 images, corresponds to group of images that seem to have rubble and part of broken house in common. Looking at their annotations, Group 1 texts, a set of mutual words presents itself, for example: '*damage, death, earthquake, rise*'. Images in column Group 2 Images exhibit broken buildings that are somehow still intact. Pairs of words that are mutual are found as well, such as '*earthquake, damage, lay waste to*'. It is noteworthy that these 2 sets of images where chosen based on their appearance rather than on their annotations. The fact that the annotations are almost similar highlights the fact that people use the same set of words when describing some visually similar attachments.

Table 11 includes some examples of false positives. Unlike the false negatives, these images do not share some evident features, but they do however share some similarities with '*Infrastructure Damage*' images. Some of these similarities are presented through the images in column Group 1 images. These images represent '*Nature Damage*' in an urban environment, with houses and roads. The semantics in column Group 1 Texts have mutual words as well, mainly '*damage, storm, cause'*. Images in column Group 2 Images contain nature-like scenarios, which are a bit different from standard images presented in Figure 2. The term '*earthquake*' is shared amongst them.

Table 11 False Positives of Best Augmented Classifier

Group 1 Images	Group 1 texts	Group 2 Images	Group 2 texts
	Damage, storm		Aid, country, earthquake
	Damage, earthquake, cause, hit, magnitude, severe, storm		Earthquake, hit
	Damage, cause, more		damage
	Damage, report, storm		Kill, earthquake, hit, seashore, magnitude, least

4. Low-Level Gaussian Kernel SVM vs Two-score Linear SVM

In this section, we address how the misclassifications of low-level Gaussian kernel SVM differ from Two-score Linear SVM. We do so first by comparing the amount of false images from the best low-level classification obtained via *Gabor-Gist-Gray-HSV*, *Classifier 1-Visual*, with the false images from the best *two-score learning* obtained by augmented *Gabor-Gist-Gray-HSV* visual features with *Data01*, Classifier 1-Augmented. We also compare the false images from the best *two-score learning* classification, obtained from *Gray-Gist-HSV-Data01*, *Classifier 2-Augmented*, with the false images from the low-level classifier of the corresponding visual feature, *Classifier 2-Visual*. Misclassifications of *Classifier Data01* are also taken into consideration. Table 12 indicates the number of false positives and negatives of above mentioned classifiers.

Table 12 Low-Level Misclassifications vs Two-Score Learning Misclassifications

Classifier	False Positives	False Negatives
Classifier Data01	205	22
Classifier 1-Visual	105	21
Classifier 1-Augmented	25	77
Classifier 2-Visual	108	22
Classifier 2-Augmented	25	77

Notice that in both cases, the number of false positives drops significantly, whereas the number of false negatives increases as the semantic information is augmented with the low-level features. This can be attributed to the ambiguity of the annotations used to describe the images. As addressed in the Semantic Kernel Classification section, several filtered annotations are assigned to both classes. It would appear that some positive observations, represented by the new feature vector consisting of the scores from the low-level and semantic classifiers, are being grouped with the negative observations, as a result of the optimal separation plane, considering this is a linear SVM. In both cases however, the *Accuracy* increases with the Two-score learning. Some errors remain even after Two-score learning. For both Classifier 1 and Classifier 2, the same observations remained misclassified after two-score learning, shown in Table 13. Although there is no apparent visual indication that necessitates this confusion in the classifier, there may be a hidden feature that is shared amongst the classes, to which these observations are misclassified.

Table 13 Observations that remained misclassified as False Negatives

	
earthquake	earthquake, aid, survivor

Table 14 Observations that remained misclassified as False Positives

				
damage, storm	damage, storm	damage, cause, storm	damage, cause, more	damage, report, storm
				
damage, earthquake, cause, hit,	aid, country, earthquake	empty	empty	

The surviving false negatives are both located in a forest-like area. The affected places in both images are the houses themselves, nonetheless, the holistic scene in the image represents nature. The annotations of the false positives are not distinct, yet share some keywords such as damage, storm and empty. Since we rely on annotations to improve the performance, it appears that these annotations do not help in this distinction, since they may be misclassified originally. A deeper look at the annotations is shown in Table 15.

Table 15 Annotations of Misclassified Observations

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, report, storm	0	3	0	0
damage, storm	10	8	8	0
damage, cause, More	1	1	1	1
aid, country, earthquake	0	1	1	0
damage, earthquake, cause, hit, magnitude, severe, storm	0	2	2	0
Nothing	57	24	24	0

The first two rows correspond to the false negative observations. The phrase in the first row is correctly classified every time, while the term 'earthquake' fails to be recognized as a true negative since the majority of the observations made of this phrase are labeled as true positives. This fact should contribute to the improvement of the augmented results, so failure of improvement can be attributed to the value representing the image. For the false positives, we can say that the observations failed to be correctly classified when augmented since the phrases associated with them are also misclassified, except for the first two phrases in light blue, that had no false positives during phrase classification.

5. CA vs Strong CA

Similar to Table 12, we take a look on the misclassifications that occurred after implementing Strong CA. Table 16 highlights the number of false positives and negatives from both implementations

Table 16 Misclassification Values of CA and Strong CA

Approach	False Positives	False Negatives
Raw CA	177	181
Raw Strong CA	164	20
Visual Features CA	62	108
Visual Features Strong CA	58	78

From the numbers mentioned above, we deduce that *Strong CA* is better than regular CA, where in both cases, visual features and raw pixel clustering, the number of false positives and false negatives decreased. However, some of the originally misclassified images remained misclassified after the clustering with the modified algorithm. A sample of the images are presented in Table 17.

Table 17 Sample of Surviving Misclassification of Raw Clustering

False Positives		False Negatives	
			
			

By inspecting the images in Table 17, we notice that the false positives contain pixel values (colors) that are usually present in an urban environment, mainly the grey color. False negative on the other hand, albeit contain grey, have also brown and green present in them, making it hard for the clustering to recognize as an infrastructure damage image.

Table 18 Sample of Surviving Misclassification of Visual Features

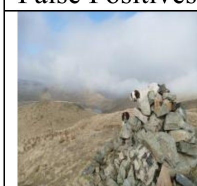
False Positives		False Negatives	
			
			

Table 18 includes a sample of the misclassifications carried on after using *Strong* CA on visual features. These images definitely contain similar features with the misclassified class in terms of shape, texture, or color.

6. Remarks

The misclassified images support the following conclusion, ‘damage’ in itself does not a have a proper visual identification. The damage in the images may be that of nature, i.e. fallen tree, mudslide, etc., nonetheless some images are being identified as infrastructure damage. A house may be broken or wall destroyed, but if it is present in natural scene, some are being identified as nature damage. This assumption may stem from the fact that ‘damage’ in itself is not a fixed term, and has no fixed visual representation. 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 as describing fallen building or a chopped tree. With no proper definition of damage, classification remains tricky. Tricky in a sense that features can’t definitely 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, we used features from various spaces, to try to properly model an image, and ultimately understanding the type of damage present in it.

CHAPTER VI

CONCLUSION

Natural disasters leave behind chaos and disruption. This leaves first responders and organizations overwhelmed with pressuring decisions based on uncertain variables which is the result of poor assessment of the types of damage that occurred. In this paper, we present a novel hybrid approach in humanitarian computing. We propose a new disaster-related categorization for scene classification: 'types of damage' with two classes: '*Infrastructure*' and '*nature*' damage. This helps relieve involved parties from tedious activities such as manually checking the news and social media for a proper damage assessment, which helps create a better response. To test our theory, we created a database based on information gathered from Twitter. These tweets are initially manually labeled via crowdsourcing. In our current framework, the observations are represented using both low-level features and semantic attributes. The proposed semantic processing scheme includes grouping the words based on the mutual words found in their WordNet synsets. A separation technique is proposed to keep closely connected words together, splitting the large groups into smaller ones based on the joining words frequency. The annotations are then projected on several bag-of-words to obtain binary semantic descriptors. After testing several feature combinations, the best visual features A*ccuracy* was reached using *Gabor-Gist-Gray-HSV* features, *90.65%*, which is a combination of shape, color, and texture features. The best semantic features A*ccuracy* was reached using *Data07*, the semantic descriptor created by projecting the annotations on the bag of words formed via the 7th WordNet synsets, with Gaussian Kernel SVM, *83.53%*. The *Accuracy* improved to *92.24%* when two-score learning utilized the decision and score from *Data01* and *Gist-Gray-HSV*, to train a new Linear SVM. The complete bag of words used for creating *Data01* and *Data07* can be found in the appendix. Although the increase in *Accuracy* is not significant, the performance in general, *Precision, Sensitivity, and Recall,* are all improved once visual and semantic features are augmented together. A modified cortical algorithm is also created that is more biologically plausible, were we build on the concept of strength encoding from the anatomy. In this modified version, a strong input leads to a higher frequency of spikes at the output of the neuron. The clustering *Accuracy* showed an increase against the regular CA on both visual and semantic levels, which can be attributed to the more human-like model of the neuron. The work presented in this paper can be considered the first attempt to identify types of damage, and most importantly figuring out the best way to visually represent it.

Future work includes moving further into sub-class understanding, such as buildings or road damage for built-infrastructure, and/or hydrological or landslide under nature damage. Additionally, we will take a step back to look into damage/non-damage classification. This will eventually facilitate damage image indexing and retrieval. In terms of image representation, other visual features can be investigated. For semantic processing, future steps might include leveraging prior knowledge of prominent words in forms of weights, including negation, and also pursuing other representations that serve to differentiate between the same phrases being labeled as both true positives and true negatives. The proposed semantic approach can be also employed to different languages as long as the bag of words are properly defined. Online learning can also be explored by updating the weights of the classifier based on the decision and score of the incoming observations, forcing this approach to adapt with emerging tweets.

APPENDIX

I. Bag of Words corresponding to *Data01:*

1. rise,rises,ascent,acclivity,raise,climb,upgrade,climbs,lift,arise,move up,go up,come up,uprise,risen,rising,ascension,wage hike,hike,wage increase,salary increase,raises
2. death,decease,expiry,perish,exit,pass away,expire,kick the bucket,cash in one's chips,buy the farm,conk,give-up the ghost,drop dead,pop off,choke,croak,snuff it,died,deaths
3. photograph,photo,exposure,pic,photos,movie,film,moving picture,moving-picture show,motion picture,motion-picture show,picture show,flick,pics
4. kill,killed,putting to death,kills,toss off,pop,bolt down,belt down,pour down,down,drink down,downed
5. home,place,homes,family,household,menage,families,topographic point,spot,property
6. aid,assist,assistance,help,care,attention,tending,helps,assisting
7. lay waste to,waste,devastate,desolate,ravage,scourge,devastating,devastates,devastated
8. country,state,land,area,areas,dry land,ground,solid ground,terra firma
9. damage,harm,impairment,injury,hurt,trauma,injuries,damaged
10. report,describe,account,reported,study,written report,reports
11. earthquake,quake,temblor,seism,earthquakes,quakes
12. cause,do,make,caused,causes,done
13. hit,hits,impinge on,run into,collide with,struck
14. injure,wound,injured,injures
15. More,Thomas More,Sir Thomas More,more
16. seashore,coast,seacoast,sea-coast
17. survivor,subsister,survivors
18. strike,work stoppage,strikes
19. year,twelvemonth,yr
20. deadly,lifelessly
21. magnitude
22. toll
23. people
24. least
25. dead
26. powerful
27. severe,terrible,wicked,awful,dire,direful,dread,dreaded,dreadful,fearful,fearsome,frightening,horrendous,horrific
28. wind,air current,current of air,winds
29. storm,violent storm,storms
30. landslide,landslides
31. tree,trees
32. nature

II. Bag of Words corresponding to *Data07:*

1. lift,rise,rises,originate,arise,develop,uprise,spring up,grow,risen,wax,mount,climb,surface,come up,rise up,rising,go up,climbing,pilfer,cabbage,purloin,pinch,abstract,snarf,swipe,hook,sneak,filch,nobble,lifted
2. bring,get,convey,fetch,induce,stimulate,cause,have,make,causes,getting,experience,receive,undergo,got,taken,took,lend,impart,bestow,contribute,gets
3. family,household,house,home,menage,homes,kin,kinsperson,families,houses,place
4. hit,hits,strike,impinge on,run into,collide with,strikes
5. kill,killed,kills,stamp out,killing
6. help,aid,helping,helps
7. absolutely,perfectly,utterly,dead
8. death,deaths
9. deadly,mortal
10. walk out,struck
11. earthquake
12. damage
13. quake
14. magnitude
15. toll
16. people
17. least
18. more
19. powerful
20. coast
21. year
22. survivors
23. fart,farting,flatus,wind,breaking wind,winds
24. video,picture,pictures,depict,render,showing
25. tree,shoetree,trees
26. storm,storms
27. nature
28. caused
29. reported
30. severe

BIBILIOGRAPHY

Abel, Fabian, et al. "Semantics Filtering Search= Twitcident. Exploring Information in Social Web Streams." Proceedings of the 23rd ACM Conference on Hypertext and Social Media.

Alqhtani, Samar M., Suhuai Luo, and Brian Regan. "Fusing Text and Image for Event Detection in Twitter." arXiv preprint arXiv:1503.03920 (2015).

"Arab Spring to cost Middle East $800 bln, HSBC estimates." Reuters. 09/09/2013 2013. <http://www.reuters.com/article/mideast-economy-politics-idUSL6N0HZ2IT20131009>.

Ashktorab, Zahra, et al. "Tweedr: Mining Twitter to Inform Disaster Response." Proc.of ISCRAM (2014).

Banks, Grant. "A.N.T. Aid Necessities Transporter concept vehicle." 01/11/2016 2011. <http://newatlas.com/ant-aid-necessities-transporter-concept-vehicle/17402/>.

Berg, Tamara L., Alexander C. Berg and Jonathan Shih. "Automatic Attribute Discovery and Characterization from Noisy Web Data." European Conference on Computer Vision.

Bosch, Anna, Andrew Zisserman and Xavier Muñoz. "Scene Classification Via pLSA." European Conference on Computer Vision.

Caragea, Cornelia, et al. "Classifying Text Messages for the Haiti Earthquake." Proceedings of the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM2011).

Castillo, Carlos. Big Crisis Data: Social Media in Disasters and Time-Critical Situations. Cambridge University Press, 2014.

Charalampopoulos, Ioannis, and Ioannis Anagnostopoulos. "A Comparable Study Employing Weka clustering/classification Algorithms for Web Page Classification." Informatics (PCI), 2011 15th Panhellenic Conference on.

Chen, Mengen, Xiaoming Jin and Dou Shen. "Short Text Classification Improved by Learning Multi-Granularity Topics." IJCAI.

Chou, Y-Y, and Linda G. Shapiro. "A Hierarchical Multiple Classifier Learning Algorithm." Pattern Analysis & Applications 6.2 (2003): 150-68.

Cresci, Stefano, et al. "A Linguistically-Driven Approach to Cross-Event Damage Assessment of Natural Disasters from Social Media Messages." Proceedings of the 24th International Conference on World Wide Web.

Dalal, Navneet, and Bill Triggs. "Histograms of Oriented Gradients for Human Detection." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).

Dalton, Lori, Virginia Ballarin, and Marcel Brun. "Clustering Algorithms: On Learning, Validation, Performance, and Applications to Genomics." Current Genomics 10.6 (2009): 430-45.

Debarati, Guha-Sapir, PH Hoyois, and R. Below. Annual Disaster Statistical Review 2013: The Numbers and Trends. Brussels: CRED, 2014.

Deschacht, Koen, Marie-Francine Moens and Wouter Robeyns. "Cross-Media Entity Recognition in nearly Parallel Visual and Textual Documents." Large Scale Semantic Access to Content (Text, Image, Video, and Sound).

Duan, Kun, et al. "Discovering Localized Attributes for Fine-Grained Recognition." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.

Edelman, G. M., and V. B. Mountcastle. "The Mindful Brain." MIT Press (1982).

Elberrichi, Zakaria, Abdellatif Rahmoun, and Mohamed Amine Bentaallah. "Using WordNet for Text Categorization." Int.Arab J.Inf.Technol. 5.1 (2008): 16-24.

"Engineering for Everyone Tackles Disaster Relief." 2013. <https://cee.umass.edu/news/engineering-everyone-tackles-disaster-relief>.

Felzenszwalb, Pedro F., and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation." International Journal of Computer Vision 59.2 (2004): 167-81.

Feng, Yansong, and Mirella Lapata. "Automatic Image Annotation using Auxiliary Text Information." ACL.

Ferrari, Vittorio, and Andrew Zisserman. "Learning Visual Attributes." Advances in Neural Information Processing Systems.

Gorman, Jim. "5 Natural Disasters Headed for the United States." 09/05/2006 2006. <http://www.popularmechanics.com/science/environment/a4875/3852052/>.

Hajj, Nadine, Yara Rizk, and Mariette Awad. "A MapReduce Cortical Algorithms Implementation for Unsupervised Learning of Big Data." Procedia Computer Science 53 (2015): 327-34.

Holmes, Geoffrey, Andrew Donkin and Ian H. Witten. "Weka: A Machine Learning Workbench." Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on.

Hu, Jian, et al. "Enhancing Text Clustering by Leveraging Wikipedia Semantics." Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Huang, Jing, S. Ravi Kumar and Ramin Zabih. "An Automatic Hierarchical Image Classification Scheme." Proceedings of the Sixth ACM International Conference on Multimedia.

"Humanitarian Engineering and Computing." 2016. <http://www.coventry.ac.uk/study-at-coventry/faculties-and-schools/engineering-environment-and-computing/humanitarian-engineering-and-computing/>.

Imran, Muhammad, et al. "Processing Social Media Messages in Mass Emergency: A Survey." ACM Computing Surveys (CSUR) 47.4 (2015): 67.

Imran, Muhammad, et al. "AIDR: Artificial Intelligence for Disaster Response." Proceedings of the 23rd International Conference on World Wide Web.

Imran, Muhammad, et al. "Coordinating Human and Machine Intelligence to Classify Microblog Communications in Crises." Proc.of ISCRAM (2014).

Imran, Muhammad, and Carlos Castillo. "Towards a Data-Driven Approach to Identify Crisis-Related Topics in Social Media Streams." Proceedings of the 24th International Conference on World Wide Web.

Imran, Muhammad, et al. "Extracting Information Nuggets from Disaster-Related Messages in Social Media." Proc.of ISCRAM, Baden-Baden, Germany (2013).

"Internet Users." 2016. <http://www.internetlivestats.com/internet-users/#sources>.

Kamps, Jaap, et al. Words with Attitude. Citeseer, 2001.

Kanoun, Karim, et al. "Low Power and Scalable Many-Core Architecture for Big-Data Stream Computing." 2014 IEEE Computer Society Annual Symposium on VLSI.

Lampert, Christoph H., Hannes Nickisch and Stefan Harmeling. "Learning to Detect Unseen Object Classes by between-Class Attribute Transfer." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.

Lindsay, Bruce R. Social media and disasters: Current uses, future options, and policy considerations (2011).

Liu, Hugo, and Push Singh. "ConceptNet—a Practical Commonsense Reasoning Tool-Kit." BT technology journal 22.4 (2004): 211-26.

"LuminAid." 2016. <https://luminaid.com/pages/how-it-works>.

Ma, Anna, et al. "Improving Image Clustering using Sparse Text and the Wisdom of the Crowds." 2014 48th Asilomar Conference on Signals, Systems and Computers.

Mahoney, Matt. "About the Test Data." 2011. <http://mattmahoney.net/dc/textdata.html>.

Mao, Jianchang, and Anil K. Jain. "Texture Classification and Segmentation using Multiresolution Simultaneous Autoregressive Models." Pattern Recognition 25.2 (1992): 173-88.

Mikolov, Tomas, et al. "Efficient Estimation of Word Representations in Vector Space." arXiv preprint arXiv:1301.3781 (2013).

Mileti, Dennis. Disasters by Design: A Reassessment of Natural Hazards in the United States. Joseph Henry Press, 1999.

Miller, George A., et al. "Wordnet: A Lexical Database for the English Language. 2002." Available from:< cogsci.princeton.edu/wn (2013).

Mojsilović, Aleksandra, José Gomes, and Bernice Rogowitz. "Semantic-Friendly Indexing and Quering of Images Based on the Extraction of the Objective Semantic Cues." International Journal of Computer Vision 56.1-2 (2004): 79-107.

Okolloh, Ory. "Ushahidi, Or 'testimony': Web 2.0 Tools for Crowdsourcing Crisis Information." Participatory learning and action 59.1 (2009): 65-70.

Oliva, Aude, and Antonio Torralba. "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope." International journal of computer vision 42.3 (2001): 145-75.

Phan, Xuan-Hieu, Le-Minh Nguyen and Susumu Horiguchi. "Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-Scale Data Collections." Proceedings of the 17th International Conference on World Wide Web.

Poria, Soujanya, et al. "Fusing Audio, Visual and Textual Clues for Sentiment Analysis from Multimodal Content." Neurocomputing 174 (2016): 50-9.

Purohit, Hemant, and Amit P. Sheth. "Twitris v3: From Citizen Sensing to Analysis, Coordination and Action." ICWSM.

Riebeek, Holli. "The Rising Cost of Natural Hazards." 03/28/2005 2005.

Sahami, Mehran, and Timothy D. Heilman. "A Web-Based Kernel Function for Measuring the Similarity of Short Text Snippets." Proceedings of the 15th International Conference on World Wide Web.

Saleemi, Imran, Lance Hartung and Mubarak Shah. "Scene Understanding by Statistical Modeling of Motion Patterns." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.

Shi, Jianbo, and Carlo Tomasi. "Good Features to Track." Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on.

Toppa, Sabrina. "The Baltimore Riots Cost an Estimated $9 Million in Damages." 05/14/2015 2015. <http://time.com/3858181/baltimore-riots-damages-businesses-homes-freddie-gray/>.

Torresani, Lorenzo, Martin Szummer and Andrew Fitzgibbon. "Efficient Object Category Recognition using Classemes." European Conference on Computer Vision.

Tsai, Grace, et al. "Real-Time Indoor Scene Understanding using Bayesian Filtering with Motion Cues." 2011 International Conference on Computer Vision.

"Twitter." 2016. <https://about.twitter.com/company>.

"United Nations Disaster Assessment and Coordination." <http://www.unocha.org/what-we-do/coordination-tools/undac/overview>.

"The United Nations Office for Disaster Risk Reduction." <http://www.unisdr.org/who-we-are/mandate>.

Vailaya, Aditya, et al. "Image Classification for Content-Based Indexing." IEEE Transactions on Image Processing 10.1 (2001): 117-30.

Vailaya, Aditya, Anil Jain, and Hong Jiang Zhang. "On Image Classification: City Images Vs. Landscapes." Pattern Recognition 31.12 (1998): 1921-35.

Van de Weijer, Joost, Cordelia Schmid and Jakob Verbeek. "Learning Color Names from Real-World Images." 2007 IEEE Conference on Computer Vision and Pattern Recognition.

Vogel, Julia, and Bernt Schiele. "Semantic Modeling of Natural Scenes for Content-Based Image Retrieval." International Journal of Computer Vision 72.2 (2007): 133-57.

Wang, Gang, and David Forsyth. "Joint Learning of Visual Attributes, Object Classes and Visual Saliency." 2009 IEEE 12th International Conference on Computer Vision.

Wang, Huayan, Stephen Gould, and Daphne Roller. "Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding." Communications of the ACM 56.4 (2013): 92-9.

Wang, Pu, and Carlotta Domeniconi. "Building Semantic Kernels for Text Classification using Wikipedia." Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Withnall, Adam. "World peace? These are the only 11 countries in the world that are actually free from conflict." 08/14/2014 2014. <http://www.independent.co.uk/news/world/politics/world-peace-these-are-the-only-11-countries-in-the-world-that-are-actually-free-from-conflict-9669623.html>.

Xiao, Jianxiong, et al. "Sun Database: Large-Scale Scene Recognition from Abbey to Zoo." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.

Yanai, Keiji, and Kobus Barnard. "Image Region Entropy: A Measure of Visualness of Web Images Associated with One Concept." Proceedings of the 13th Annual ACM International Conference on Multimedia.

Yih, Wen-Tau, and Christopher Meek. "Improving Similarity Measures for Short Segments of Text." AAAI.

Zelikovitz, Sarah, and Haym Hirsh. "Improving Short Text Classification using Unlabeled Background Knowledge to Assess Document Similarity." Proceedings of the Seventeenth International Conference on Machine Learning.