Disaster Image Classification Using Capsule Networks

Soudabeh Taghian Dinani
Department of Computer Science
Kansas State University
Manhattan, KS, USA
soudabehtaghian@ksu.edu

Doina Caragea
Department of Computer Science
Kansas State University
Manhattan, KS, USA
dcaragea@ksu.edu

***Abstract*—When a disaster happens, affected individuals may use social media platforms, such as Twitter or Facebook, to ask for help or post information about the disaster. From a disaster response point of view, it is important to filter posts, in particular, text and images that provide situational awareness information, in a timely manner. For image classification, capsule networks have shown superiority over convolutional neural networks (CNN). Given their success in other application domains, in this study, we used capsule networks to classify disaster images as Informative or Non-informative. Using publicly available images collected from several disasters, we compared capsule network models with ResNet-18 models, for both in-domain and cross-domain settings. The results showed that the capsule network models had better performance for all the disaster datasets considered in the in-domain experiments, and also for most of the cross-domain pairs of disasters used in the study.**

***Index Terms*—Disaster images, image classification, convolutional neural networks, ResNet-18, capsule networks (CapsNets).**

I. Introduction

Social media has a great impact on our daily lives, and has become an important source of information in the recent years. When a disaster (such as a tsunami, earthquake, flood, hurricane) strikes a place, many people use social media as a medium of communication, given that traditional emergency phone services may become unavailable due to a large volume of calls [1]. Therefore, social media is of great importance in disaster response and management, and has been ranked the fourth among popular sources for acquiring emergency information [2]. Affected people in disaster zones can use social media (e.g., Twitter) to ask for help or to share emergency information [2]. This information should be filtered in a timely manner, so that disaster planners and responders can use it to improve their response operations and allocation of resources.

There have been many studies on the use of deep learning approaches for filtering useful situational information from tweets. Some studies have focused on getting information from either texts [3, 4, 5] or images [6, 7, 8], while others have considered both text and images together [9, 10]. For training a supervised deep learning model, a large number of data points is needed. However, in the case of a disaster, manually labeling data can be expensive and time consuming, and thus the number of data points is limited (especially in the early hours of a disaster). This problem can be somewhat alleviated by using data augmentation techniques and pre-trained models [11], or sometimes using labeled data from previous disasters, together with domain adaptation techniques, to train classifiers for a disaster of interest [5, 7]. It has been shown that this approach is much more useful if the previous disaster is of different nature to the current one [7].

To complement previous approaches that address the limitations posed by the scarcity of the labeled data, we propose to use capsule networks for classifying disaster images posted during disasters as Informative or Non-informative. One of the main properties of capsule networks is the property of equivariance, which captures the spatial relationship between features [12]. Jiménez-Sánchez *et. al.* [13] used capsule networks to analyze medical images and showed that the equivariance property of capsule networks translates to models that can be effectively trained with a smaller number of labeled data points as compared to the standard convolutional neural networks. As an added benefit, they also showed that capsule networks have the ability to handle imbalanced datasets. As disaster image datasets are both small and imbalanced, capsule networks represent an attractive approach for disaster image classification. Therefore, this work presents a case study of capsule networks to disaster image classification.

Given this context, our key contributions are as follows:

- We used CapsNet models to classify disaster images. To the best of our knowledge, our study is the first to use CapsNets for disaster image classification.
- We conducted in-domain experiments and compared CapsNet models with CNN models to investigate how the CapNets handle smaller amounts of labeled data, as compared to the standard CNN models. Specifically, we used CapsNets with ResNet-18 as a backbone, and compared them with ResNet-18 baselines. The experiments were run on seven different crisis datasets.
- We also conducted cross-domain experiments using both CapsNet and ResNet-18. The purpose of these experiments was to study the ability of the CapsNet models to transfer knowledge from a prior source domain to the target domain, as compared to the ResNet-18 baselines.

We thank the National Science Foundation and Amazon Web Services for support from grant IIS-1741345, which enabled the research and the computation in this study.

We ran experiments on a variety of source-target disaster pairs (formed using the seven crisis datasets from the in-domain setting).

The rest of the paper is organized as follows: We describe related work in Section II, and provide background and approaches, including a brief description of ResNet-18 architecture and Capsule Networks, in Section III. We introduce the experimental setup in Section IV, while presenting and discussing the results in Section V. Finally, we conclude the paper and provide ideas for future work in Section VI.

II. Related Work

Works related to this paper fall into two categories, specifically works on disaster image classification and works on capsule networks, as discussed below.

A. Disaster image classification

Many studies have used social media data to identify useful information for disaster response [1]. Early studies have focused on text analysis. However, more recent studies have also addressed the analysis of social media images, as images have been shown to contain useful situational awareness information that complements or adds to the information available in text [14]. Specifically, social media images provide detailed on-site information from the perspective of the eyewitnesses of the disaster [15], and can serve as an ancillary, yet rich source of visual information in disaster response.

Several image and multi-modal image/text datasets have been published [6, 16, 17], and contributed to advances in the area of image classification for disaster response. Nguyen *et. al.* [6] used CNNs to perform damage assessment, while Li *et. al.* [7] used domain adversarial neural networks (DANN) to identify damage images. Agarwal *et. al.* [18] proposed a multi-modal framework, called Crisis-DIAS, which uses both textual and visual information from tweets. Madichetty and Sridevi [19] also used both image and text features to classify crisis-related tweets. In the method proposed in this paper, a combination of a CNN and a standard Artificial Neural Network (ANN) was used for the text-based classification component of the model, while the fine-tuned VGG-16 architecture was used for the image-based classification component of the model. Afterwards, the late fusion technique was used to combine the output of these two models to determine if the tweet label is Informative or Non-informative. In [20], both the semantic text and image features are combined to classify a multi-modal tweet post. Abavisani *et. al.* [10] also used a multi-modal fusion approach, in order to detect crisis events for three different tasks based on the combination of information in both images and text.

As can be seen, all these works and others that are focused on image analysis in the disaster domain have used different variants of CNN models to process images. As opposed to that, in our study, we propose to use capsule networks, which have been designed to address fundamental limitations of CNNs. In particular, we focus on the ability of the CapsNets to handle smaller labeled datasets and study them in both in-domain and cross-domain settings.

B. Capsule Networks

CapsNets [21, 22] use viewpoint invariant representations to address a fundamental limitation of CNNs, specifically, the fact that CNNs do not capture spatial relationships between features. Given their superior performance as compared to CNNs, CapsNets have been recently used in many application domains. For example, CapsNets have been used in natural language processing applications, such as text classification [23], and multi-label text classification and question answering [24]. They have also been used successfully in computer vision. For instance, [25] used CapsNets for biomedical image segmentation. Jiménez-Sánchez *et. al.* [13] showed that CapsNet can overcome the challenges in the medical image classification domain, where datasets are usually small and imbalanced. Afshar *et. al.* [26] proposed a CapsNet-based framework, called COVID-CAPS, to diagnose COVID-19 cases based on X-ray images. Similarly, [27] proposed a CapsNet model, called Detail-Oriented Capsule Networks (DECAPS), to automatically classify COVID-19 patients from Computed Tomography (CT) scans, while [28] used CapsNets to detect pneumonia from Chest X-ray images. Singh *et. al.* [29] used a CapsNet-based approach to create a high resolution image out of a very low resolution (VLR) image, and demonstrated the feasibility of the approach for VLR digit and face recognition.

Given the good performance of CapsNets in other application domains, especially for image classification tasks, our goal is to study the usefulness of CapsNets for disaster image classification, a task for which the available datasets are relatively small and imbalanced. To the best of our knowledge, this is the first time CapsNets are used in the disaster domain.

III. Background and Approach

CapsNets have a CNN backbone. We use ResNet-18 as the backbone of the CapsNet models in this study, and also as a baseline when evaluating the CapsNet models. Thus, in this section, we first review the ResNet-18 architecture, and then describe the CapsNet model that we used.

A. ResNet-18 Architecture

The architecture of CNN has three different types of layers including convolution layers (together with non-linear ReLU activations), sub-sampling layers (a.k.a., pooling layers), and classification layers, with the first two types of layers present in the lower levels of the network, and classification layers at the top of the network. Specifically, the architecture of a CNN consists of a sequence of convolutional layers, interspersed with max-pooling layers, followed by fully connected (FC) layers, and finally a softmax classification layer. The outputs of the convolution and max-pooling layers can be seen as feature maps. Lower level layers correspond to more general feature maps, while the upper level layers correspond to more specific feature maps. The feature maps associated with the

last convolution/max-pooling layers are provided as inputs to the classification layers, and a prediction is made based on the scores produced by the final softmax layer. In the last fully connected layer, the class with the highest score obtained from a softmax layer will be chosen as the predicted class [30].

CNN architectures have gained a lot of popularity in computer vision, due to the considerable improvements that they have produced in this field. In particular, ResNet [31] is a popular CNN architecture, which won the ImageNet image classification competition in 2015. The advantage of the ResNet architecture over other CNN architectures is that it incorporates identity shortcuts, which help prevent the problem of performance saturation and/or degradation, generally faced when training deeper networks. By skipping some layers, this identity shortcut connection considers the layer residual mapping instead of only the original mapping [32]. We used ResNet-18 (which has 18 layers) as the backbone of the CapsNet models in this paper. Given that the data used in our experiments has only two classes, the FC layer of the pre-trained ResNet-18 was changed from 1000 to 2 classes.

B. Capsule Networks

CapsNets were first proposed by Hinton *et. al.* [21, 22]. The main motivation for CapsNets development was given by a fundamental problem faced by standard CNNs, specifically, the fact that they are invariant to translation (due to the pooling operation), but cannot learn the spatial relationships between features. Thus, a CNN model can wrongly identify an image containing two eyes and a nose as a face, even when these features are not in the right location. Furthermore, the pooling operation results in loss of information. To compensate for this loss, CNNs need a large training dataset [12]. For example, in the case of face detection, the CNN model needs to be trained on images of faces taken from a variety of viewpoints. On the other hand, CapsNets are equivariant. For an image that shows a rotated face, a CapsNet will output the probability of the image to be classified as a face, along with the rotation degree. Therefore, CapsNets can be trained with smaller datasets and can generalize better to new unseen images with viewpoints different from those in the training set. A brief description of CapsNets is given in the following paragraph.

A CapsNet consists of several layers, with many capsules in each layer. A capsule encapsulates a group of neurons in itself. The output of each capsule can encode features of an entity (e.g., its pose, its probability of existence, its deformation, etc.) in the image. The active capsules at a lower level (the children of the higher-level capsules) vote for the capsules of the higher level (the possible parents). This is done based on transformation matrices, representing viewpoint-invariant relationship between the child capsule and the parent capsule. These transformation matrices are trained, and they change with a change in the viewpoint, making the viewpoint between the child capsule (part) and the parent capsule (whole or object) invariant. Votes are accumulated, and if several children capsules agree on a parent capsule, that capsule becomes active. This process, called routing-by-agreement, is repeated several times. The routing is done in such a way that ensures that one child capsule is routed to only one parent capsule. The result of the routing is a parsing tree with a hierarchical structure [21, 22, 33]. Dynamic routing [21] and EM-routing [22] are two routing algorithms (with some differences) that implement this mechanism.

Tsai *et. al.* [33] proposed a new routing algorithm, called "Inverted Dot-Product Attention Routing", which is similar to the inverted attention mechanism. In this algorithm, the parent capsules try to get attention from the children capsules. The agreement between the previous iteration vote of the parent capsule and the current iteration vote of the children capsules is determined by the routing likelihood. There are two other changes in this routing algorithm: 1) the authors used a Layer Normalization; and 2) they used concurrent iterative routing instead of sequential iterative routing; therefore, the states of the capsules and the routing likelihoods are simultaneously inferred (not in a layer-wise fashion). These changes help to scale up the model [33].

The CapsNet architecture used in this work was adapted from [33] and is presented in Figure 1. In the original architecture [33], either several ResNet computational blocks or a single convolutional layer were used as the backbone. As opposed to that, we used ResNet-18 as our CapsNet's backbone to benefit from the pre-training of the ResNet-18 on the ImageNet dataset. For an image of size 224×224 pixels, the size of the image output by ResNet-18 is 7×7. Therefore, after the image is fed to the backbone ResNet-18, the output will be upsampled by the factor of 2 before being sent to the convolutional layer, which is followed by LayerNorm, primary capsules, two convolutional capsule layers, and finally two fully-connected capsule layers. Given that we have only two classes, the number of class capsules were changed to 2 in our model. The class probability is obtained by applying the softmax function to the resulting logits. The class with higher probability is chosen as the predicted class. As can be seen in the architecture, the routing algorithm is used from the primary capsules all the way to the class capsules [33].

IV. Experimental Setup

In this section, the dataset used in the experiments, along with setup, evaluation metrics, and baseline are presented. The aim of the designed experiments is to answer to the following research questions:

- Does adding the capsule layers to the backbone ResNet-18 improve the performance for disaster image classification in in-domain and cross-domain settings?
- How does the performance of CapsNets on smaller datasets compare to the performance on larger datasets?
- In the cross-domain setting, how does the ability of the CapsNet to transfer information between source and target disasters compare for disasters of the same type versus disasters of different types?

To answer these questions, we selected a diverse dataset that allows us to simulate a variety of scenarios. The details about

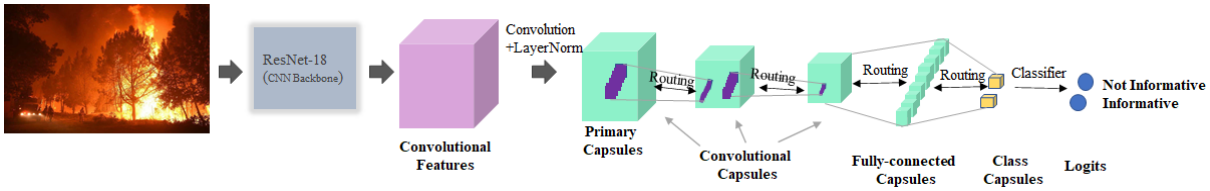


Fig. 1: Overview of the capsule network (CapsNet) architecture, adapted from [33]

TABLE I: Data distribution of the CrisisMMD dataset for the task of identifying Informative versus Non-informative images.

Dataset	Event	Not-inf.	Inf.	Total	Year
D0	California Fire	604	984	1588	2017
D1	Hurricane Harvey	1982	2461	4443	2017
D2	Hurricane Irma	2303	2222	4525	2017
D3	Hurricane Maria	2330	2232	4562	2017
D4	Iraq Iran Earthquake	200	400	600	2017
D5	Mexico Earthquake	541	841	1382	2017
D6	Sri Lanka Floods	773	252	1025	2017

the dataset, the experimental setup, evaluation metrics and baseline models are provided in the next subsections.

A. Datasets

The dataset used in this study is called CrisisMMD [16]. This dataset was assembled by collecting tweets during seven natural disasters including, Hurricane Irma, Hurricane Harvey, Hurricane Maria, California Wildfires, Mexico Earthquake, Iraq-Iran Earthquake, and Sri Lanka Floods. The dataset includes both images and texts related to the corresponding disasters; however, we only used the images of the dataset in this study. Furthermore, while the dataset was originally designed to address several image classification tasks, we only focus on the task of classifying images in two classes: Informative and Non-informative. The detailed data distributions of the disasters in the dataset (the number of Informative and Non-informative images in each disaster) are presented in Table I.

B. Setup

For the in-domain experiments, the dataset was randomly split into three parts: 70% as training set, 10% as the development set and 20% as the test set. Each experiment was run three times on each dataset and the average over runs were reported for each metric. For the cross-domain experiments, all the data for the source domain was used for the training, while the development set and testing set of target domain were used for hyper-parameter selection and evaluation, respectively. In order to perform a fair comparison with the in-domain results, the development and test sets are the same as those used in the in-domain experiments. Also, for the cross-domain experiments, we formed pairs of disasters based on the chronological order in which the disasters in the CrisisMMD dataset occurred. This order from the first disaster to the last one is: D6, D1, D2, D3, D5, D0, D4. Given the varying sizes of the specific disaster datasets, for some disaster pairs, the source dataset has more samples than the target dataset, while for others, the source has fewer samples, or a comparable number of samples. Furthermore, the CrisisMMD dataset consists of disasters of both similar and different types, which enables us to experiment with source and target disasters of the same type (for example, in one experiments the source data is Hurricane Harvey and the target data is Hurricane Irma), as well as disasters of different types (for example, in one experiment the source data is Sri Lanka Floods and the target data is California Fires). Thus, the diversity of pairs formed with disasters from CrisisMMD allows us to study CapsNets under a variety of scenarios.

Each experiment, both in the in-domain and cross-domain settings, was first run on the ResNet-18 baseline (pre-trained on ImageNet). All the layers, except for Conv-5 group and FC layer, were frozen. Parameters of Conv-5 were initialized with the pre-trained values and fine-tuned for each experiment, while parameters of the FC layer were randomly initialized and trained from scratch. To identify a good set of hyper-parameters, we performed extensive tuning on the Hurricane Maria dataset, as this dataset is large compared to other datasets in CrisisMMD and it is also relatively balanced. The final hyper-parameters used in the experiments are: a learning rate of 5e-07, a weight decay (corresponding to L2 regularization) of 0.1, a batch size of 32 images, and a max number of epochs of 70. The exact number of epochs for an experiment (leading to the best model) was identified separately based on the development set in that experiment. The final performance for an experiment was evaluated using the best model on the test set corresponding to that experiment. The input images were resized to 224×224 (as the ResNet-18 model was pre-trained on images of this size).

For each experiment with the CapsNet model, the ResNet-18 model (which was tuned on the disaster dataset for the corresponding experiment in the previous part) was used as the backbone. This backbone was frozen, while the other layers of the CapsNet model were trained. The hyper-parameters selected for this part (using fine-tuning on Hurricane Maria as before) are: a learning rate of 1e-05, a weight decay of 0 (i.e., no L2 regularization), dropout with a rate of 0.3, a batch size of 32 images, and a max epoch number of 70. Similar to ResNet-18, each experiment is run for 70 epochs, but the number of epochs that result in the best accuracy on the development set is selected and used to evaluate performance on the test set. The input images for CapsNet were also resized to 224×224. The matrix format was used for the pose of each capsule, and the number of routings was 2.

The same data augmentation approach is used for the

TABLE II: Classification results for in-domain experiments: Precision (Pr.), Recall (Re.) and F1-measure (F1). For each metric, the best values in a row are bold-faced.

	Resnet-18			CapsNet		
Data	Pr.	Re.	F1	Pr.	Re.	F1
D0	79.343	78.616	78.599	**83.642**	**83.543**	**83.515**
D1	77.296	77.290	77.150	**79.014**	**78.979**	**78.843**
D2	74.473	74.401	74.393	**75.256**	**75.212**	**75.188**
D3	75.731	75.731	75.724	**77.503**	**77.449**	**77.419**
D4	79.044	71.389	71.046	**82.599**	**82.778**	**82.210**
D5	76.217	75.362	75.400	**78.122**	**78.261**	**78.146**
D6	79.042	75.447	74.636	**83.850**	**84.390**	**83.735**

training set of all the experiments. Firstly, the image is resized to 225 × 225, then it is center-cropped to the size of 224 × 224. Then, padding of four is used and the image is randomly cropped to the size of 224 × 224. Afterwards, images are flipped randomly in the horizontal direction.

C. Evaluation Metrics and Baseline Models

The metrics used are weighted precision, recall and F1-measure. We chose to focus on these metrics as some of the datasets are class imbalanced (i.e., the ratio between the number of instances in the two classes is high).

As mentioned before, the baseline used in our study is ResNet-18. Thus, the results of the CapsNet models are compared with the results of the ResNet-18 models. The disaster-tuned ResNet-18 is also used as the backbone in the CapsNet architecture, which enables us to assess the ability of the capsule layers (added after ResNet-18) to improve the performance of the baseline models.

V. Experimental Results

The results of the in-domain experiments and cross-domain experiments are shown in Tables II and IV, respectively. The reported results are averaged over the three replicas of each experiment. We discuss the results with respect to our research questions, and present a brief error analysis in what follows.

A. Discussion of the Results

Our first research question was focused on the ability of the capsule layers, added to the backbone ResNet-18 layers, to improve the performance on disaster image classification. As Table II shows, for all the in-domain experiments, the results of the CapsNet model are better than those of the baseline for all evaluation metrics considered. Furthermore, Table IV shows that the results of the CapsNet model are also better than those of the baseline models for most cross-domain experiments. Specifically, the results obtained with CapsNet for three metrics considered (precision, recall and F1 score) are better than the ones of the ResNet-18 in 18 out of 21 cases (corresponding to different source-target combinations formed based on the chronological order). This suggests that CapsNets are capable of improving the performance on the task of classifying disaster images, as compared ResNet-18.

TABLE III: Difference between the CapsNet and ResNet-18 mean F1-score values for in-domain experiments, and percentage change (%F1 Change). The disasters with the smallest datasets and highest observed differences are bold-faced. The Order column shows the rank of the dataset in terms of size, while the Ratio column shows the class ratio.

Data	Order	Size	Ratio	F1	% F1 Change
D4	1	600	2.000	**11.164**	**15.713%**
D6	2	1025	0.326	**9.099**	**12.191%**
D5	3	1382	1.555	**2.746**	**3.641%**
D0	4	1588	1.629	**4.916**	**6.254%**
D1	5	4443	1.242	1.693	2.194%
D2	6	4525	0.965	0.795	1.068%
D3	7	4562	0.958	1.695	2.238%

Our second research question was focused on the benefits of the CapsNet model as compared with the ResNet-18 baseline, when smaller or larger datasets are used for training. To facilitate the analysis of the results with respect to the dataset size, Table III shows the differences between the CapsNet and ResNet-18 mean values for the F1 metric for the in-domain experiments (with the differences for the smaller datasets shown in bold font). More specifically, each difference is calculated as the mean F1 value of CapsNet minus the corresponding F1 mean value for of ResNet-18, i.e., $F1(CapsNet) - F1(ResNet)$, according to Table III. In addition to the difference, we also show the percentage change as $[F1(CapsNet) - F1(ResNet)] \div F1(ResNet)] \times 100$. The order of the datasets based on their size from the smallest to the largest is: D4, D6, D5, D0, D1, D2, and D3, as shown in Table III. The table also shows the size of each dataset, and the ratio of Informative to Non-informative samples, as the class imbalance can influence the results as well. As can be seen in Table III, the largest difference (with 15,713% increase in F1-score) is for D4, which is the smallest dataset (600 samples). The second largest improvement in the F1-score is observed for D6 (with 12.191% increase in F1-score), which is the next smallest dataset (1025 samples). Besides being the smallest, the two datasets also exhibit the largest class imbalance, with Informative to Non-informative ratio being 2:1 in D4, and approximately 1:3 in D6. The third and fourth largest differences are observed for D0 and D5, respectively, which are the next datasets according to size (with D5 containing 1382 samples, and D0 containing 1588 samples). While D5 is smaller than D0, D0 has a slightly higher class imbalance, and overall the improvement seen from the CapsNet is larger for D0 as compared to D5. For the larger datasets, D1, D2, D3, which have larger size (approximately, 4500 samples each) and relatively balanced class ratio, the percent increases are in the range 1-2%. Together, these results suggest that both the dataset size and the class imbalance ratio contribute to the improved performance of CapsNet. These results are consistent with the findings in prior work [13], in which it was shown that the CapsNets has a better generalization ability (compared to CNN) in the case of small size datasets and class-imbalance. This behavior can be attributed to the equivariant

TABLE IV: Classification results for the cross-domain experiments: precision (Pr.), recall (Re.) and F1-measure (F1). For each metric, the best results in a row are bold-faced.

		ResNet-18			CapsNet		
S	T	Pr.	Re.	F1	Pr.	Re.	F1
D1	D0	72.452	71.908	72.038	**73.894**	**73.166**	**73.324**
D2	D0	69.102	66.981	67.317	**72.450**	**69.392**	**69.818**
D3	D0	66.048	61.006	60.708	**69.981**	**67.400**	**67.850**
D5	D0	62.958	59.434	57.730	**66.049**	**66.352**	**64.541**
D6	D0	63.477	50.105	42.170	**65.344**	**50.734**	**47.36**
D6	D1	69.142	62.087	59.344	**74.001**	**68.919**	**68.291**
D1	D2	72.390	71.786	71.656	**73.789**	**73.738**	**73.736**
D6	D2	63.324	58.858	54.394	**66.038**	**62.431**	**59.836**
D1	D3	74.016	73.794	73.768	**75.241**	**75.183**	**75.176**
D2	D3	74.769	74.708	74.704	**76.368**	**76.279**	**76.276**
D6	D3	67.019	63.596	61.284	**69.405**	**66.228**	**64.499**
D0	D4	75.541	70.833	70.963	**78.995**	**76.389**	**76.923**
D1	D4	**78.809**	**77.778**	**77.860**	78.120	76.667	77.079
D2	D4	74.543	73.333	73.422	**79.142**	**75.000**	**75.652**
D3	D4	77.451	75.556	75.739	**78.460**	**75.833**	**76.412**
D5	D4	**83.714**	80.833	81.120	83.392	**82.778**	**82.954**
D6	D4	76.337	61.944	60.480	**77.717**	**63.333**	**63.527**
D1	D5	75.151	74.638	74.800	**75.405**	**75.362**	**75.349**
D2	D5	75.059	**74.155**	**74.383**	**75.623**	73.671	73.950
D3	D5	74.489	73.430	73.660	**76.268**	**73.913**	**74.150**
D6	D5	76.026	67.874	67.014	**76.623**	**68.357**	**68.124**

TABLE V: Difference between the CapsNet and ResNet-18 mean F1-score values for cross-domain experiments, and percentage change (%F1 Change). The highest observed differences are bold-faced.

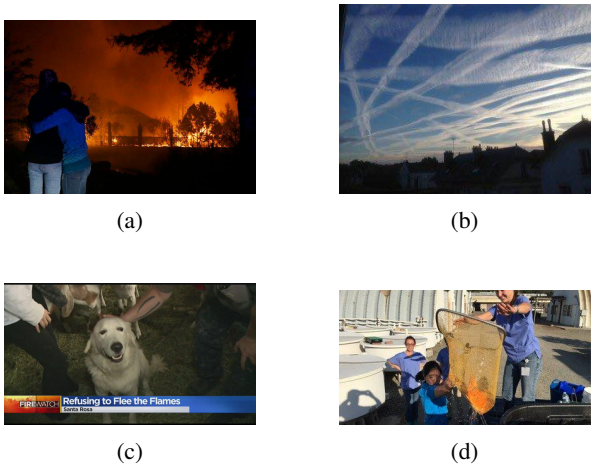
Source	Target	F1	%F1 Change
D1 (Hurricane)	D0 (Fires)	1.286	1.785%
D2 (Hurricane)	D0 (Fires)	2.501	3.715%
D3 (Hurricane)	D0 (Fires)	**7.142**	**11.765%**
D5 (Earthquake)	D0 (Fires)	**6.811**	**11.798%**
D6 (Floods)	D0 (Fires)	**5.190**	**12.307%**
D6 (Floods)	D1 (Hurricane)	**8.947**	**15.077%**
D1 (Hurricane)	D2 (Hurricane)	2.080	2.903%
D6 (Floods)	D2 (Hurricane)	**5.442**	**10.005%**
D1 (Hurricane)	D3 (Hurricane)	1.408	1.909%
D2 (Hurricane)	D3 (Hurricane)	1.572	2.104%
D6 (Floods)	D3 (Hurricane)	3.215	5.246%
D0 (Fires)	D4 (Earthquake)	**5.960**	**8.399%**
D1 (Hurricane)	D4 (Earthquake)	-0.781	-1.003%
D2 (Hurricane)	D4 (Earthquake)	2.230	3.037%
D3 (Hurricane)	D4 (Earthquake)	0.673	0.889%
D5 (Earthquake)	D4 (Earthquake)	1.834	2.261%
D6 (Floods)	D4 (Earthquake)	3.047	5.038%
D1 (Hurricane)	D5 (Earthquake)	0.549	0.734%
D2 (Hurricane)	D5 (Earthquake)	-0.433	-0.582%
D3 (Hurricane)	D5 (Earthquake)	0.490	0.665%
D6 (Floods)	D5 (Earthquake)	1.110	1.656%

property of CapsNets that can complement for limited data, and generalize better to unseen images as compared to CNN baselines. However, a more controlled study, where only one of these characteristics is varied at a time, is needed to better understand the effect of each characteristic.

Our last question was related to the ability of the CapsNet model to transfer information from the source to the target, in the cross-domain setting, when the source and target disasters are of similar types or of different types, respectively. To answer this question, for each pair of disasters considered in our cross-domain experiments, Table V shows the differences between the CapsNet F1-score and the corresponding ResNet-18 F1-score (and also the percent change in F1-score). Six pairs with the largest differences and percentage change are bold-faced (the percentage change for these pairs varies from 8.399% to 15.005%). All these six cases correspond to experiments where the source and target disasters are of different types, although we want to emphasize that not all pairs with different types of disasters show a big difference. To better interpret the results, we consider cases with the same target and different sources. For the experiments with D0 (California Fires) as the target, all the sources are of types different than D0. Three of these experiments are among the ones with the highest F1 difference and percentage change. In particular, when the source is D3 (Hurricane Maria), D5 (Mexico Earthquake), or D6 (Sri Lanka Floods), there is an increase of 11.765%, 11.798%, and 12.307% in F1, respectively. When D2 (Hurricane Irma) is the target, the improvement in F1 is higher when D6 (Sri Lanka Floods) is the source (10.005% increase in F1) than when D1 (Hurricane Harvey) is the source (2.903% increase in F1). We observe similar results when D3 (Hurricane Maria) is the target. For the experiments with D4 (Iraq-Iran Earthquake) as the target, the two largest increases in F1 are for the cases with D0 (California Fires) and D6 (Sri Lanka Floods) as the sources. However, two other experiments with sources D3 (Hurricane Maria) and D1 (Hurricane Harvey) of types different than D4 do not have larger increase in F1 as compared to the experiment where the source is D5 (Mexico Earthquake). Given that the size of the training dataset was observed to affect the improvements obtained with CapsNets, the results here suggest that the size of the source may also be a factor that determines the improvement, in addition to the nature of the disasters. In particular, using D5 (Mexico Earthquake) and D6 (Sri Lanka Floods) as sources generally leads to the largest increases within a group (corresponding to a target). These are the two smallest datasets in our set, supporting the hypothesis that source size is important in addition to source type. More experiments are needed to tease out the individual factors involved here.

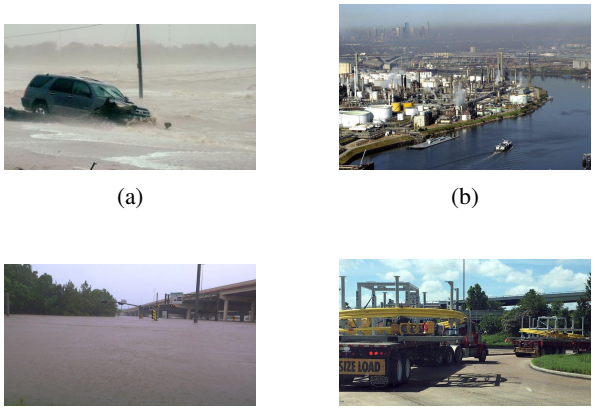
B. Error Analysis

Figure 2 (in-domain setting) and Figure 3 (cross-domain setting) show examples of images that are correctly classified by one of the models (i.e., CapsNet or ResNet-18), while they are mis-classified by the other model. The in-domain models used to classify the images in Figure 2 are trained and tested on California Fires (D0). Specifically, Figure 2 shows examples of Informative and Non-informative images correctly classified by CapsNet and mis-classified by ResNet-18 in the top row (a) and (b), and examples of Informative and Non-informative images correctly classified by ResNet and mis-classified by CapsNet in the bottom row (b) and (c). Given that the capsule network has the property of being equivariant to viewpoint, it correctly identifies image (a) as Informative with respect to California Fires, and image (b) as Non-informative. However, it seems to mistakenly classify image (d) as Informative,



(a) (b) (c) (d)

Fig. 2: Error Analysis for the in-domain experiment focused on the D0 (California Fire) dataset. (a) Example of an Informative image that is correctly classified by CapsNet, while miss-classified by ResNet-18; (b) Example of a Non-informative image that is correctly classified by CapsNet, and miss-classified by ResNet-18; (c) Example of an Informative image that is correctly classified by ResNet-18, and miss-classified by CapsNet; (d) Example of a Non-informative image correctly classified by ResNet-18, and miss-classified by CapsNet.



(a) (b) (c) (d)

Fig. 3: Error Analysis for the cross-domain experiment where D6 (Sri Lanka Floods) dataset is used as source and D1 (Hurricane Harvey) as target. (a) Example of an Informative image that is correctly classified by CapsNet, while miss-classified by ResNet-18; (b) Example of a Non-informative image that is correctly classified by CapsNet, and miss-classified by ResNet-18; (c) Example of an Informative image correctly classified by ResNet-18, and miss-classified by CapsNet; (d) Example of a Non-informative image correctly classified by ResNet-18 and miss-classified by CapsNet.

possibly because it identifies an orange region that resembles fire. On the other hand, it mis-classifies image (c) as Non-informative as the fire shown in that image is flattened into the image heading, while the main view of the image does not show anything related to fire. All together, the two models agree on 271 test instances (168 Informative and 103 Non-informative), and disagree on 47 instances (29 Informative and 18 Non-informative). The CapsNet model predicts correctly 33 of the images where the models disagree, while ResNet predicts correctly only 14 of those instances.

The cross-domain models used to classify the images in Figure 3 are trained using Srilanka Floods (D6) as source, and tested on Hurricane Harvey (D1) as target. CapsNet correctly classifies images (a) and (b) as Informative and Non-informative, respectively, with respect to Hurricane Harvey. However, it mistakenly classifies image (c) as Non-informative, possibly because the flooded road could look as a regular road in some viewpoint, and it mis-classifies image (d) as Informative, possibly because it learns to associates big machines with recovery from a hurricane. In this experiment, the two models agree on 682 instances (334 Informative and 348 Non-Informative), and they disagree on 206 instances (158 Informative and 48 Non-informative). When in disagreement, the CapsNet model is correct in 159 cases, while the ResNet-18 model is correct in only 47 cases.

VI. Conclusions and Future Work

In this paper, CapsNets were used with the purpose of classifying disaster images as Informative or Non-informative. Extensive in-domain and cross-domain experiments were carried out to investigate the ability of the CapsNets for this task. The results suggest that the CapsNet model has a better performance compared to the baseline ResNet-18 for all the disaster datasets for in-domain experiments. It also showed an improvement in the evaluation metrics for most of the considered cross-domain pairs of disasters. Most importantly, the results showed that the CapsNet models could improve the performance of the ResNet-18 baseline when the size of the training datasets is small, or the datasets are imbalanced.

For future work, firstly, in order to better understand the effect of sample size and imbalance, we plan to conduct controlled experiments, where only one of these factors is varied at a time. Secondly, we intend to use CapsNet models for other classification tasks with several classes (e.g. Disaster Types). Moreover, we plan to investigate if benefits of the CapsNets hold when a deeper ResNet architecture is used as the baseline model and the backbone. Finally, we plan to make use of the texts in the tweets to design multi-modal CapsNet models for classifying disaster posts as sometimes the information in an image and the corresponding text is complementary and using both may result in better performance.

References

[1] A. Saroj and S. Pal, "Use of social media in crisis management: A survey," *International Journal of Disaster Risk Reduction*, p. 101584, 2020.

[2] J. Kim and M. Hastak, "Social network analysis: Characteristics of online social networks after a disaster," *International Journal of Information Management*, vol. 38, no. 1, pp. 86–96, 2018.

[3] C. Caragea, A. Silvescu, and A. H. Tapia, "Identifying informative messages in disaster events using convolutional neural networks," in *ISCRAM*, 2016, pp. 137–147.

[4] D. T. Nguyen, S. Joty, M. Imran, H. Sajjad, and P. Mitra, "Applications of online deep learning for crisis response using social media information," *arXiv preprint arXiv:1610.01030*, 2016.

[5] F. Alam, S. Joty, and M. Imran, "Domain adaptation with adversarial training and graph embeddings," *arXiv preprint arXiv:1805.05151*, 2018.

[6] D. T. Nguyen, F. Ofli, M. Imran, and P. Mitra, "Damage assessment from social media imagery data during disasters," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. ACM, 2017, pp. 569–576.

[7] X. Li, D. Caragea, C. Caragea, M. Imran, and F. Ofli, "Identifying disaster damage images using a domain adaptation approach." in *ISCRAM*, 2019.

[8] M. Imran, F. Alam, U. Qazi, S. Peterson, and F. Ofli, "Rapid damage assessment using social media images by combining human and machine intelligence," *arXiv preprint arXiv:2004.06675*, 2020.

[9] F. Ofli, F. Alam, and M. Imran, "Analysis of social media data using multimodal deep learning for disaster response," *arXiv preprint arXiv:2004.11838*, 2020.

[10] M. Abavisani, L. Wu, S. Hu, J. Tetreault, and A. Jaimes, "Multimodal categorization of crisis events in social media," in *CVPR*, 2020, pp. 14 679–14 689.

[11] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019.

[12] M. K. Patrick, A. F. Adekoya, A. A. Mighty, and B. Y. Edward, "Capsule networks–a survey," *Journal of King Saud University-Computer and Inf.. Sciences*, 2019.

[13] A. Jiménez-Sánchez, S. Albarqouni, and D. Mateus, "Capsule networks against medical imaging data challenges," in *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Springer, 2018.

[14] M. Imran, F. Ofli, D. Caragea, and A. Torralba, "Using ai and social media multimodal content for disaster response and management: Opportunities, challenges, and future directions," 2020.

[15] M. Bica, L. Palen, and C. Bopp, "Visual representations of disaster," in *CSCW*, 2017, pp. 1262–1276.

[16] F. Alam, F. Ofli, and M. Imran, "Crisismmd: Multimodal twitter datasets from natural disasters," in *ICWSM*, Stanford, California, USA, 2018.

[17] H. Mouzannar, Y. Rizk, and M. Awad, "Damage identification in social media posts using multimodal deep learning," in *ISCRAM 2018*, Rochester, NY, 2018.

[18] M. Agarwal, M. Leekha, R. Sawhney, and R. R. Shah, "Crisis-dias: Towards multimodal damage analysis-deployment, challenges and assessment," in *AAAI*, vol. 34, no. 01, 2020, pp. 346–353.

[19] S. Madichetty and M. Sridevi, "Classifying informative and non-informative tweets from the twitter by adapting image features during disaster," *Multimedia Tools and Applications*, vol. 79, no. 39, pp. 28 901–28 923, 2020.

[20] G. Nalluru, R. Pandey, and H. Purohit, "Relevancy classification of multimodal social media streams for emergency services," in *SMARTCOMP*, 2019.

[21] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[22] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," in *ICLR*, 2018.

[23] J. Kim, S. Jang, S. Choi, and E. Park, "Text classification using capsules. arxiv p," *arXiv preprint arXiv:1808.03976*, 2018.

[24] W. Zhao, H. Peng, S. Eger, E. Cambria, and M. Yang, "Towards scalable and reliable capsule networks for challenging nlp applications," *arXiv preprint arXiv:1906.02829*, 2019.

[25] R. LaLonde, Z. Xu, I. Irmakci, S. Jain, and U. Bagci, "Capsules for biomedical image segmentation," *Medical Image Analysis*, vol. 68, p. 101889, 2020.

[26] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images," *arXiv preprint arXiv:2004.02696*, 2020.

[27] A. Mobiny, P. A. Cicalese, S. Zare, P. Yuan, M. Abavisani, C. C. Wu, J. Ahuja, P. M. de Groot, and H. Van Nguyen, "Radiologist-level covid-19 detection using ct scans with detail-oriented capsule networks," *arXiv preprint arXiv:2004.07407*, 2020.

[28] A. Mittal, D. Kumar, M. Mittal, T. Saba, I. Abunadi, A. Rehman, and S. Roy, "Detecting pneumonia using convolutions and dynamic capsule routing for chest x-ray images," *Sensors*, vol. 20, no. 4, p. 1068, 2020.

[29] M. Singh, S. Nagpal, R. Singh, and M. Vatsa, "Dual directed capsule network for very low resolution image recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 340–349.

[30] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," *arXiv preprint arXiv:1803.01164*, 2018.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[32] C. Kwan, B. Chou, J. Yang, and T. Tran, "Compressive object tracking and classification using deep learning for infrared videos," in *Pattern Recognition and Tracking XXX*, vol. 10995, 2019, p. 1099506.

[33] Y.-H. H. Tsai, N. Srivastava, H. Goh, and R. Salakhutdinov, "Capsules with inverted dot-product attention routing," *arXiv preprint arXiv:2002.04764*, 2020.