Robust Training of Social Media Image Classification Models

Firoj Alam[1], Tanvirul Alam[2], Ferda Ofli[1], Muhammad Imran[1]
[1]Qatar Computing Research Institute, HBKU, Doha, Qatar
[2]BJIT Limited, Dhaka, Bangladesh
[1]{fialam, fofli, mimran}@hbku.edu.qa, [2]tanvirul.alam@bjitgroup.com

***Abstract*—Images shared on social media help crisis managers gain situational awareness and assess incurred damages, among other response tasks. As the volume and velocity of such content are typically high, real-time image classification has become an urgent need for a faster disaster response. Recent advances in computer vision and deep neural networks have enabled the development of models for image classification for a number of tasks, including detecting crisis incidents, filtering irrelevant images, classifying images into specific humanitarian categories, and assessing the severity of the damage. To develop robust models, it is necessary to understand the capability of the publicly available pre-trained models for these tasks, which remains to be under-explored in the crisis informatics literature. In this study, we address such limitations by investigating ten different network architectures for four different tasks using the largest publicly available datasets for these tasks. We also explore various data augmentation strategies, semi-supervised techniques, and a multitask learning setup. In our extensive experiments, we achieve promising results.**

***Index Terms*—Social media image classification, Multitask Learning, Crisis informatics, Humanitarian tasks, Disaster response**

I. Introduction

Social media is widely used during natural or human-induced disasters to disseminate information and obtain valuable insights quickly. People post content (i.e., through different modalities such as text, image, and video) on social media to ask for help, to offer support, to identify urgent needs, or to share their feelings. Such information is helpful for humanitarian organizations to plan and launch relief operations. As the volume and velocity of the content are significantly high, it is crucial to have systems to process social media content to facilitate rapid response automatically. There has been a surge of research studies in this domain in the past couple of years. The focus has been to analyze social media data and develop computational models using varying modalities to extract actionable information. Among different modalities (e.g., text and image), more focus has been given to textual content analysis compared to imagery content (see [1]–[3] for comprehensive surveys). However, many past research works have demonstrated that images shared on social media during a disaster event can also assist humanitarian organizations. For example, Nguyen et al. [4] use images shared on Twitter to assess the severity of the infrastructure damage, and Mouzannar et al. [5] focus on identifying damages in infrastructure as well as environmental elements.

For a clear understanding, we provide an example pipeline in Figure 1a which demonstrates how different disaster-related image classification models can be used in real-time for information categorization. As presented in the figure, the four different classification tasks such as *(i)* disaster types, *(ii)* informativeness, *(iii)* humanitarian, and *(iv)* damage severity assessment, can significantly help crisis responders during disaster events. For example, disaster type classification model can be used for real-time event detection as shown in Figure 1b. Similarly, the informativeness model can be used to filter non-informative images, the humanitarian model can be used to discover fine-grained categories, and the damage severity model can be used to assess the impact of the disaster. Current literature reports either one or two tasks using one or two network architectures. Another limitation is that there have been limited datasets for disaster-related image classification. Very recently, the study by Alam et al. [6] developed a *benchmark dataset*,[1] which is consolidated from existing publicly available resources. The development process of this dataset consists of data curation from different existing sources, development of new data for new tasks, creating non-overlapping[2] training, development, and test sets. The reported benchmark dataset targeted the four tasks as shown in Figure 1a.

In this study, we build upon [6] and address the aforementioned limitations by posing the following Research Questions (RQs):

- **RQ1:** Can data consolidation help?
- **RQ2:** Among various neural network architectures with pre-trained weights, which one is more suitable for different downstream disaster-related image classification tasks?
- **RQ3:** Does data augmentation or semi-supervised learning help to improve the model performance?
- **RQ4:** Is multitask learning an ideal solution to reduce computational complexity when there is need to make predictions for multiple tasks simultaneously?

To understand the benefits of data consolidation (*RQ1*), we extended the work by Alam et al. [6] with more in-depth analysis. Our motivation for *RQ2* is that there has been significant progress in neural network architectures for image processing

[1]We refer to this dataset as *Crisis Benchmark Dataset* throughout the paper.
[2]Duplicate images are identified between test and training sets and moved from the test set to the training set.

(a) Disaster image classification pipeline.

(b) Event detection use case showing landslide images.

Fig. 1: Disaster image classification pipeline that demonstrate a real use case – landslide image classification.

in the last few years; however, they have not been widely explored in the *crisis informatics*[3] domain for disaster response tasks. Hence, we investigated several neural network architectures for different disaster-related image classification tasks. Since augmentation and self-training-based techniques [7], [8] have shown success to yield more generalized models and sometimes improve the performance, we pose *RQ3* and investigate them for the mentioned tasks. For the social media image classification tasks shown in Figure 1, it is necessary to run the mentioned models in sequence or parallel for the same input image. Running multiple models can be prohibitively expensive when there is a need to analyze many social media images. Having a single model for dealing with multiple tasks can significantly alleviate the computational complexity. Hence, we pose *RQ4* to instigate research in this direction. The *Crisis Benchmark Dataset* has not been originally developed for multitask learning setup. However, the related metadata information (e.g., image ids) are available, and we utilized such information to create data splits for multitask learning while trying to maintain the same training, development, and test splits. As our experiment shows, this is challenging due to the incomplete labels for different tasks (see more details in Section IV-F).

To summarize, our contributions in this study are as follows:

- We present more detailed results highlighting the benefit of data consolidation.
- We address four tasks using several state-of-the-art neural network architectures on different data splits.
- We investigate various data augmentation techniques and show that model generalization improves with data augmentation.
- We explore semi-supervised learning and multitask learning to have a single model while addressing multiple tasks. Based on the findings, we provide research directions for future studies.
- We also provide insights using Gradient-weighted Class Activation Mapping [9] to demonstrate what class-specific discriminative properties are learned by the networks.

The rest of the paper is organized as follows. Section II provides a brief overview of the existing work. Section III introduces the tasks and describes the datasets used in this

[3]https://en.wikipedia.org/wiki/Disaster_informatics

study. Section IV explains the experiments, Section V presents the results, and Section VI provides a discussion. Finally, we conclude the paper in Section VIII.

II. Related Work

A. Social Media Content for Disaster Response

Most of the earlier research efforts in crisis informatics are mainly focused on textual content analysis [3]. However, lately there has been a growing interest on the imagery content analysis as images posted on social media during disasters can play significant role as reported in many studies [4], [10]–[16]. Recent works include categorizing the severity of damage into discrete levels [4], [12], [13] or quantifying the damage severity as a continuous-valued index [17], [18]. Such models were also used in real-time disaster response scenarios by engaging with emergency responders [19].

The studies on image processing in the crisis informatics domain are relatively few compared to the studies on analyzing textual content for humanitarian aid.[4] With recent successes of deep learning for image classification, research works have started to use social media images for humanitarian aid. The importance of imagery content on social media for disaster response tasks has been reported in many studies [4], [10]–[13], [20], [21]. For instance, the analysis of flood images has been studied in [10], in which the authors reported that the existence of images with the relevant textual content is more informative. Similarly, the study by Daly and Thom [11] analyzed fire event images, which are extracted from social media data. Their findings suggest that images with geotagged information are helpful to locate the fire-affected areas.

The analysis of imagery content shared on social media has recently been explored using deep learning techniques for damage assessment purposes. Most of these studies categorize the severity of damage into discrete levels [4], [12], [13] whereas others quantify the damage severity as a continuous-valued index [17], [18]. Other related work include data scarcity issue by employing more sophisticated models such as adversarial networks [22], [23], disaster image retrieval [24], image classification in the context of bush fire emergency [25], flooding photo screening system [26], sentiment analysis from disaster image [27], monitoring natural disasters using satellite images [28], and flood detection using visual features [29].

B. Real-time Systems

Recently, Alam et al. [21] presented an image processing pipeline to extract meaningful information from social media images during a crisis situation, which has been developed using deep learning-based techniques. Their image processing pipeline includes collecting images, removing duplicates, filtering irrelevant images, and finally classifying them with damage severity. Such a system has been used during several disaster events, and one such example is the deployment during Hurricane Dorian, reported in [19]. The system has been deployed for 13 days, and it collected around ∼280K images. These images are then automatically classified and used by a volunteer response organization, Montgomery County Maryland Community Emergency Response Team (MCCERT). Another example use case is the early detection of disaster-related damage to cultural heritage [30].

[4]https://en.wikipedia.org/wiki/Humanitarian_aid

C. Multimodality (Image and Text)

The exploration of multimodality has also received attention in the research community [31], [32]. In [31], authors explore different fusion strategies for multimodal learning. Similarly, in [32] a cross-attention-based network is exploited for multimodal fusion. The study in [33] reports a multimodal system for flood image detection, which achieves a precision of 87.4% in a balance test set. In another study, the authors propose a similar multimodal system for on-topic vs. off-topic social media post classification and report an accuracy of 92.94% with imagery content [34]. The study in [35] explores different classical machine learning algorithms to classify relevant vs. irrelevant tweets using textual and imagery information. On the imagery content, they achieved an F1-score of 87.74% using XGboost [36]. The study in [37] proposes a simple, computationally inexpensive, multimodal two-stage framework to classify tweets (text and image) with built-infrastructure damage vs. nature-damage. The study investigates their approach using a home-grown dataset, and the SUN dataset [38]. Mouzannar et al. [5] proposes a multimodal dataset, which has been developed for training a damage detection model. Similarly, Ofli et al. [39] explores unimodal as well as different multimodal modeling approaches based on a collection of multimodal social media posts.

D. Transfer Learning for Image Classification

For the image classification task, transfer learning has been a popular approach, where a pre-trained neural network is used to train a model for a new task [5], [39]–[43]. For this study, we follow the same approach using different deep learning architectures. For disaster related image classification, there have been studies where transfer-learning based models have been used either as feature extractors or for fine-tuning the model. Such studies include flood detection from social media multimodal content [44], disaster related tasks in a multitask learning [45], real-time system for disaster image classification during hurricane [46], sentiment analysis from disaster images [47], aerial image classification for disaster response [48], and deep features with multimodal training [49].

E. Datasets

Currently, publicly available datasets include damage severity assessment dataset [4], CrisisMMD [50] and damage identification multimodal dataset [5]. The first dataset is only annotated for images, whereas the last two are annotated for both text and images. Other relevant datasets are Disaster Image Retrieval from Social Media (DIRSM) [51] and MediaEval 2018 [52]. The dataset reported in [53] is constructed for detecting damage as an anomaly using pre-and post-disaster images. It consists of 700,000 building annotations. A similar and relevant work is the development of the Incidents dataset

[54], which consists of 446684 manually labeled Web images with 43 incident categories. The *Crisis Benchmark Dataset* reported in [6] is the largest dataset so far for social media disaster image classification.

For this study, we use the *Crisis Benchmark Dataset*, and our study differs from [6] in a number of ways. We provide more detailed experimental results on dataset comparison (i.e., individual vs. consolidated), compare different network architectures with a statistical significance test, and report the efficacy of data augmentation. We have also utilized a large unlabeled dataset to enhance the capability of the current model. We created multitask data splits from *Crisis Benchmark Dataset* and report experimental results using both missing/incomplete and complete labels, which can serve as a baseline for future works.

III. Tasks and Datasets

For this study, we addressed four different disaster-related tasks that are important for humanitarian aid. Below we provide details of each task and the associated class labels.

A. Tasks

1) Disaster type recognition: When ingesting images from unfiltered social media streams, it is important to detect different disaster types automatically from these images. For instance, an image can depict a wildfire, flood, earthquake, hurricane, and other types of disasters. In the literature, disaster types have been defined in different hierarchical categories such as natural, human-induced, and hybrid [55]. Natural disasters are events that result from natural phenomena (e.g., fire, flood, earthquake). Human-induced disasters result from human actions (e.g., terrorist attacks, accidents, wars, and conflicts). Hybrid disasters result from human actions, which affect natural phenomena afterward (e.g., deforestation results in soil erosion and climate change). The class labels for disaster type include *(i)* earthquake, *(ii)* fire, *(iii)* flood, *(iv)* hurricane, *(v)* landslide, *(vi)* other disaster (to cover all other disaster types, e.g., plane crash), and *(vii)* not disaster (for images that do not show any identifiable disaster).

2) Informativeness: Images posted on social media during disasters do not always contain informative or useful content for humanitarian aid (e.g., an image showing damaged infrastructure due to flood, fire, or any other disaster event). It is necessary to remove any irrelevant or redundant content to facilitate crisis responders' efforts more effectively. Therefore, the purpose of this classification task is to filter out irrelevant images. The class labels for this task are *(i)* informative and *(ii)* not informative.

3) Humanitarian: An important aspect of crisis responders is to assist people based on their needs, which requires information to be classified into more fine-grained categories that can trigger specific actions. In the literature, humanitarian categories often include *affected individuals*; *injured or dead people*; *infrastructure and utility damage*; *missing or found people*; *rescue, volunteering, or donation effort*; and *vehicle damage* [50]. In this study, we focus on four categories that are deemed to be the most prominent and important for crisis responders such as *(i)* affected, injured, or dead people, *(ii)* infrastructure and utility damage, *(iii)* rescue volunteering or donation effort, and *(iv)* not humanitarian.

Fig. 2: An image annotated as *(i)* fire event, *(ii)* informative, *(iii)* infrastructure and utility damage, and *(iv)* severe damage.

4) Damage severity: Assessing the severity of the damage is important to help the affected community during disaster events. The severity of damage can be assessed based on the physical destruction of a built structure visible in an image (e.g., destruction of bridges, roads, buildings, burned houses, and forests). Following the work reported in [4], we define the categories for this classification task as *(i)* severe damage, *(ii)* mild damage, and *(iii)* little or none.

Figure 2 shows an example image with the labels for all four tasks.

B. Datasets

As mentioned earlier, we used the dataset reported in [6].[5] This dataset has been developed by consolidating existing publicly available sources, and by defining non-overlapping training, development, and test splits. For the sake of clarity and completeness, we provide a brief overview of the dataset. More details about the dataset curation and consolidation process can be found in [6].

1) Damage Assessment Dataset (DAD): The damage assessment dataset consists of labeled imagery data with damage severity levels such as severe, mild, and little-to-no damage [4]. The images have been collected from two sources: AIDR [56] and Google. To crawl data from Google, authors used the following keywords: *damage building, damage bridge, and damage road*. The images from AIDR were collected from Twitter during different disaster events such as Typhoon Ruby, Nepal Earthquake, Ecuador Earthquake, and Hurricane Matthew. The dataset contains ∼25K images annotated by paid workers as well as volunteers. In this study, we use this dataset for the informativeness and damage severity tasks. For the informativeness task, the study in [6] mapped the *mild* and *severe* images into informative class

[5]https://crisisnlp.qcri.org/crisis-image-datasets-asonam20

and manually categorized the *little-to-no damage* images into *informative* and *not informative* categories. For the damage severity task, the label *little-to-no damage* mapped into *little or none* to align with other datasets.

2) CrisisMMD: This is a multimodal (i.e., text and image) dataset, which consists of 18,082 images collected from tweets during seven disaster events crawled by the AIDR system [50]. The data is annotated by crowd workers using the Figure-Eight platform[6] for three different tasks: *(i)* informativeness with binary labels (i.e., informative vs. not informative), *(ii)* humanitarian with seven class labels (i.e., "infrastructure and utility damage", "vehicle damage", "rescue, volunteering, or donation effort", "injured or dead people", "affected individuals", "missing or found people", "other relevant information" and "not relevant"), *(iii)* damage severity assessment with three labels (i.e., severe, mild and "little or no damage"). For the humanitarian task similar class labels are grouped together. The images with labels *injured or dead people* and *affected individuals* are mapped into one class label *affected, injured, or dead people*; *infrastructure and utility damage* and *vehicle damage* are mapped into *infrastructure and utility damage*; *other relevant information*, and *not relevant* are mapped into *not humanitarian*. The images with label *missing or found people* are removed as it is difficult to identify. This results in four class labels for humanitarian task.

3) AIDR Disaster Type Dataset (AIDR-DT): AIDR-DT dataset consists of tweets collected from 17 disaster events and 3 general collections. The tweets of these collections have been collected by the AIDR system [56]. The 17 disaster events include flood, earthquake, fire, hurricane, terrorist attack, and armed-conflict. The tweets in general collections contain keywords related to natural disasters, human-induced disasters, and security incidents. Images are crawled from these collections for disaster type annotation. The labeling of these images was performed in two steps. First, a set of images were labeled as *earthquake*, *fire*, *flood*, *hurricane*, and *none of these categories*. Then, a sample of $\sim$2,200 images labeled as *none of these categories* in the previous step are selected for annotating *not disaster* and *other disaster* categories.

For the landslide category, images are crawled from Google, Bing, and Flickr using keywords landslide, mudslide, "mud slides", landslip, "rock slides", rockfall, "land slide", earthslip, rockslide, and "land collapse". As images have been collected from different sources, therefore, it resulted in having duplicates. Duplicate filtering has been applied to remove exact- and near-duplicate images to resolve this issue. Then, the remaining images were manually labeled as *landslide* and *not landslide*. The resulted annotated dataset consists of labeled images with seven categories defined in Section III-A1.

4) Damage Multimodal Dataset (DMD): The multimodal damage identification dataset consists of 5,878 images collected from Instagram and Google [5]. The authors of the study crawled the images using more than 100 hashtags, which are proposed in crisis lexicon [57]. The manually labeled data consist of six damage class labels: fires, floods, natural landscape, infrastructural, human, and non-damage. The non-damage image includes cartoons, advertisements, and images that are not relevant or useful for humanitarian tasks. The study by Alam et al. [6] re-labeled images for all four tasks: disaster type, informativeness, humanitarian, and damage severity using the same class labels discussed in the previous section.

[6]Currently acquired by https://appen.com/

C. Data Consolidation and Statistics

The datasets introduced in previous section comprises images collected from various sources such as Google, Bing, Yahoo, and Twitter. Since only the images collected from Twitter contain social media information, only those images that have Twitter's JSON objects ($\sim$27K images) have been analyzed to understand the distribution of images across different disaster events. Table I reports statistics of the collected tweets and images for different events. It appears that people share images in only 1 to 5% of the posts.

Before consolidating the datasets, each dataset has been divided into training (train), development (dev), and test sets with 70:10:20 ratio, respectively. The purpose was threefold: *(i)* train and evaluate individual datasets on each task, *(ii)* have a close-to-equal distribution from each dataset into the final consolidated dataset, and *(iii)* provide the research community an opportunity to use the splits independently. After data split, duplicate images are identified across sets and moved into the training set to create a non-overlapping test set.

For the exact- and near-duplicate image identification, we extracted feature representations for each image using a pre-trained ResNet18 [58] model and computed Euclidean distance between all possible image pairs. We then manually verified a subset of image pairs and determined a threshold of 2.6 to automatically find exact- and near-duplicate images. More details about the duplicate identification process can be found in [6].

During the experiments, the training set was used to train the model, the development set was used for the fine-tuning, and the test set was used for the final evaluation. Since the primary motivation to perform data consolidation is to develop robust deep learning models with large amounts of data, all individual training, development, and test sets are merged into the consolidated training, development, and test sets, respectively. As combining multiple datasets can results in duplicate images in train and test set, after merging the dataset, we repeat the same duplicate identification procedure to maintain non-overlapping sets for different tasks.

Finally, Tables II, III, IV, V, and VI show the label distribution of all datasets for all four tasks. Some class labels are skewed in individual datasets. For example, in disaster type datasets (Table II), the distribution of the "other disaster" label is low in the AIDR-DT dataset, whereas the distribution of the "landslide" label low in the DMD dataset. For the informativeness task, low distribution is observed for the "informative" label. Moreover, for the humanitarian task, we have low distribution for the "rescue volunteering or donation effort" label in the DMD dataset, and for the damage severity task "mild" label in CrisisMMD and DMD datasets. However, the consolidated dataset creates a fair balance across class labels for different tasks, as shown in Table VI.

TABLE I: Number of tweets and images collected during different disaster events.

Event name	Year	# Tweets	# Images	% Images	Start Date	End Date
Nepal earthquake	2015	4,223,936	132,361	3.13	25-Apr-2015	19-May-2015
Paris attack	2015	10,599,629	499,953	4.72	14-Nov-2015	3-Dec-2015
South india floods	2015	2,994,119	141,831	4.74	3-Dec-2015	6-Dec-2015
Food insecurity in Yemen	2015	1,107,931	63,686	5.75	25-Sep-2015	19-Nov-2015
Italy earthquake	2016	3,382,698	167,331	4.95	26-Oct-2016	27-Nov-2016
Hurricane Irma	2017	3,517,280	176,972	5.03	6-Sep-2017	21-Sep-2017
Hurricane Harvey	2017	6,664,349	321,435	4.82	26-Aug-2017	20-Sep-2017
Hurricane Maria	2017	2,953,322	52,231	1.77	20-Sep-2017	13-Nov-2017
Mexico earthquake	2017	383,341	7,111	1.86	20-Sep-2017	6-Oct-2017
California wildfires	2017	455,311	10,130	2.22	10-Oct-2017	27-Oct-2017
Iraq-Iran earthquake	2017	207,729	6,307	3.04	13-Nov-2017	19-Nov-2017
Sri Lanka floods	2017	41,809	2,108	5.04	31-May-2017	3-Jul-2017
Syria attacks	2017	5,381,866	107,513	2.00	6-Apr-2017	26-Apr-2017
Ukraine conflict	2017	1,268,942	30,289	2.39	5-Nov-2017	13-Nov-2017
Kerala flood	2018	3,044,703	15,767	0.52	17-Aug-2018	12-Sep-2018
Hurricane Florence	2018	623,074	12,879	2.07	11-Sep-2018	24-Sep-2018
Hurricane Michael	2018	243,263	5,106	2.10	10-Oct-2018	27-Oct-2018

TABLE II: Data split for the **disaster type** task.

Dataset	Class labels	Train	Dev	Test	Total
AIDR-DT	Earthquake	1,910	201	376	2,487
	Fire	990	105	214	1,309
	Flood	2,059	241	533	2,833
	Hurricane	1,188	142	279	1,609
	Landslide	901	119	257	1,277
	Not disaster	1,507	198	415	2,120
	Other disaster	65	6	17	88
	Total	8,620	1,012	2,091	11,723
DMD	Earthquake	130	17	35	182
	Fire	255	36	71	362
	Flood	263	35	70	368
	Hurricane	253	36	73	362
	Landslide	38	5	11	54
	Not disaster	2,108	288	575	2,971
	Other disaster	1,057	145	287	1,489
	Total	4,152	506	1,130	5,788

TABLE III: Data split for the **informativeness** task.

Dataset	Class labels	Train	Dev	Test	Total
DAD	Informative	15,329	590	2,266	18,185
	Not informative	5,950	426	1,259	7,635
	Total	21,279	1,016	3,525	25,820
CrisisMMD	Informative	7,233	635	1,507	9,375
	Not informative	6,535	551	1,621	8,707
	Total	13,768	1,186	3,128	18,082
DMD	Informative	2,071	262	573	2,906
	Not informative	2,152	240	580	2,972
	Total	4,223	502	1,153	5,878
AIDR-Info	Informative	627	66	172	865
	Not informative	6,677	598	1,796	9,071
	Total	7,304	664	1,968	9,936

TABLE IV: Data split for the **humanitarian** task.

Class labels	Train	Dev	Test	Total
CrisisMMD				
Affected, injured, or dead people	521	51	100	672
Infrastructure and utility damage	3,040	299	589	3,928
Not humanitarian	3,307	296	807	4,410
Rescue volunteering or donation effort	1,682	174	375	2,231
Total	8,550	820	1,871	11,241
DMD				
Affected, injured, or dead people	242	28	63	333
Infrastructure and utility damage	933	125	242	1,300
Not humanitarian	2,736	314	744	3,794
Rescue volunteering or donation effort	74	9	18	101
Total	3,985	476	1,067	5,528

TABLE V: Data split for the **damage severity** task.

Dataset	Class labels	Train	Dev	Test	Total
DAD	Little or none	7,881	1,101	1,566	10,548
	Mild	2,828	388	546	3,762
	Severe	9,457	673	1,380	11,510
	Total	20,166	2,162	3,492	25,820
CrisisMMD	Little or none	317	35	67	419
	Mild	547	56	125	728
	Severe	1,629	144	278	2,051
	Total	2,493	235	470	3,198
DMD	Little or none	2,874	331	778	3,983
	Mild	508	60	132	700
	Severe	857	110	228	1,195
	Total	4,239	501	1,138	5,878

IV. Experiments

Our experiments include *(i)* individual vs. consolidated dataset comparisons (*RQ1*), *(ii)* neural network architecture comparisons on the consolidated dataset (*RQ2*), *(iii)* data augmentation (*RQ3*), *(iv)* semi-supervised learning (*RQ3*), and *(iv)* multitask learning (*RQ4*). Next we first present our experimental setup, and then, discuss different experiments that we conducted in this study.

A. Experimental Setup

We employ the transfer learning approach to perform experiments, which has shown promising results for various visual recognition tasks in the literature [40]–[43]. The idea of the transfer learning approach is to use existing weights of a pre-trained model for different downstream tasks. We use

TABLE VI: Data splits for the **consolidated dataset** for all tasks.

Class labels	Train	Dev	Test	Total
Disaster Type				
Earthquake	2,058	207	404	2,669
Fire	1,270	121	280	1,671
Flood	2,336	266	599	3,201
Hurricane	1,444	175	352	1,971
Landslide	940	123	268	1,331
Not disaster	3,666	435	990	5,091
Other disaster	1,132	143	302	1,577
Total	12,846	1,470	3,195	17,511
Informativeness				
Informative	26,486	1,432	3,414	31,332
Not informative	21,700	1,622	5,063	28,385
Total	48,186	3,054	8,477	59,717
Humanitarian				
Affected, injured, or dead people	772	73	160	1,005
Infrastructure and utility damage	4,001	406	821	5,228
Not humanitarian	6,076	578	1,550	8,204
Rescue volunteering or donation effort	1,769	172	391	2,332
Total	12,618	1,229	2,922	16,769
Damage Severity				
Little or none	11,437	1,378	2,135	14,950
Mild	4,072	489	629	5,190
Severe	12,810	845	1,101	14,756
Total	28,319	2,712	3,865	34,896

the weights of the networks pre-trained using ImageNet [59] to initialize our model. We adapt the last layer (i.e., softmax layer) of the network according to the particular classification task at hand instead of the original 1,000-way classification. The transfer learning approach allows us to transfer the features and the parameters of the network from the broad domain (i.e., large-scale image classification) to the specific one. Put specifically, we design a binary classifier for the informativeness task and multi-class classifiers for the remaining three tasks. We train the models using the Adam optimizer [60] with an initial learning rate of 10^{-5}, which is decreased by a factor of 10 when accuracy on the development set stops improving for 10 epochs. The models were trained for 150 epochs. We performed all experiments using the the PyTorch library.[7] To measure the performance of each classifier, we use weighted average precision (P), recall (R), and F1-score (F1).

B. Dataset Comparisons

To determine whether consolidated data helps in achieving better performance, we train the models using training sets from the individual and consolidated datasets. However, we always test the models on the consolidated test set. As our test data is the same across different experiments, this ensures that results are comparable. Since we have four different tasks, consisting of fifteen different datasets, we only experimented with the ResNet18 [58] network architecture to manage the computational load.

[7]https://pytorch.org/

C. Network Architectures

Currently available neural network architectures come with different computational complexity. As one of our goals is to deploy the models in real-time applications, we exploit them to understand their performance differences. Another motivation is that current literature in crisis informatics only reports results using one or two network architectures (e.g., VGG16 in [39], InceptionNet in [5]), which may lead to sub-optimal outcomes. Therefore, in this study, we experiment with several neural network architectures including ResNet18, ResNet50, ResNet101 [58], AlexNet [61], VGG16 [62], DenseNet [63], SqueezeNet [64], InceptionNet [65], MobileNet [66], and EfficientNet [67].

D. Data Augmentation

Data augmentation is a commonly used technique to improve the generalization of deep neural networks in the absence of large-scale datasets. We experiment with the recently proposed RandAugment [7] method for image augmentation. In literature, RandAugment was proposed as a fast alternative for learned augmentation strategies. We used the PyTorch implementation[8] in our experiments. To increase the diversity of generated examples, we used the following 16 transformations,

1) AutoContrast
2) Equalize
3) Invert
4) Rotate
5) Color
6) Posterize
7) Solarize
8) SolarizeAdd
9) Contrast
10) Brightness
11) Sharpness
12) ShearX
13) ShearY
14) CutoutAbs
15) TranslateX
16) TranslateY

where augmentation strengths can be controlled with two tunable parameters N and M where N indicates the number of augmentation transformations to apply sequentially, and M indicates the magnitude for all the transformations.

Each transformation resides on an integer scale from 0 to 30, with 30 being the maximum strength. In our experiments, we use constant magnitude M for all augmentations. The augmentation method then boils down to randomly selecting N transformations and applying each transformation sequentially with strength corresponding to scale M.

In addition, we used *weight decay*, which is one of the most commonly used techniques for regularizing parametric machine learning models [68]. This helps to reduce the overfitting of the models and avoids exploding gradient.

We have conducted the data augmentation experiments using all ten different neural network architectures. We used a weight decay of 10^{-3} and other hyper-parameters remain the same as discussed in Section IV-A.

E. Semi-supervised Learning

State-of-the-art image classification models are often trained with a large amount of labeled data, which is prohibitively expensive to collect in many applications. Semi-supervised learning is a powerful approach to mitigate this issue and

[8]https://github.com/ildoonet/pytorch-randaugment

leverage unlabeled data to improve the performance of machine learning models. Since unlabeled data can be obtained without significant human labor, performance boost gained from semi-supervised learning comes at low cost and can be scaled easily. In literature many semi-supervised techniques has been proposed focusing on deep learning [8], [69]–[79]. Among them self-training approach is one of the earliest [80], which has been adopted for deep neural network. The self-training approach, also called pseudo-labeling [8], uses the model's prediction as a label and retrains the model against it.

For this study, we use *Noisy student* (i.e., a simple self-training approach) training, which was proposed in [69] as a semi-supervised learning approach to improve the accuracy and robustness of state-of-the-art image classification models. The algorithm consists of three main steps:

Step 1: Train a teacher model on labeled images
Step 2: Use the teacher model to generate pseudo labels on unlabeled images
Step 3: Train a student model on combined labeled and pseudo labeled images

The algorithm can be iterated multiple times by treating the student as the new teacher and labeling the unlabeled images with this model. During the learning phase of the student, different noises can be injected, such as dropout [81] and data augmentation via RandAugment [7]. The student model is made larger than or equal to the teacher. The presence of noise and larger model capacity help the student model generalize better than the teacher.

a) Labeled dataset: As for the labeled dataset, we used our consolidated datasets and ran the experiments for all tasks.

b) Unlabeled dataset: To obtain unlabeled images, we crawled images from the tweets of 20 different disaster collections (as mentioned in Section III-B3). We removed duplicates and ensured the same images are not in our labeled dataset by matching their ids and applying duplicate filtering. The resulting unlabeled dataset consists of 1,514,497 images.

c) Architecture: We ran our experiments using the EfficientNet (b1) architecture as it performed better than the other models. In addition, it is one of the models used with *Noisy student* experiments reported in [69]. One significant difference between [69] and our work is that we initialize our student model's weight with ImageNet pre-trained weights. In contrast, in [69], they train weights from scratch. Since our labeled dataset is significantly smaller than the ImageNet dataset, training from scratch substantially degrades performance in our experiments.

d) Training details: We first trained the model using the EfficientNet (b1) architecture on the labeled dataset (**Step 1**), which is referred to as the teacher model. We then predicted output for the unlabeled images (**Step 2**). After that, we trained the student EfficientNet(b1) model by combining labeled and pseudo-labeled images (**Step 3**). In this step, for the unlabeled data, we performed different filtering and balancing. We selected the images that have a confidence label greater than a certain task-specific threshold. After this, we balanced the training data so that each class has the same number of images as the class having the lowest number of images. To do this, for each class, we take the images having the highest confidence scores.

For the experiments, we used a batch size of 16 for labeled images and 48 for unlabeled images. Labeled and unlabeled images are concatenated together to compute the average cross-entropy loss. We used RandAugment with the number of augmentation, $N = 5$, and the strength of augmentation, $M = 12$. We optimized the confidence thresholds separately for different tasks using the dev sets. The thresholds for disaster types, informativeness, humanitarian, and damage severity tasks were respectively 0.7, 0.8, 0.45, and 0.45. Similar to the data augmentation experiments, we used a weight decay of 10^{-3} and kept other hyper-parameters the same as discussed in Section IV-A.

TABLE VII: Data split for multi-task setting with **incomplete/missing labels**. DS: Disaster types, Info: Informative, Hum: Humanitarian, DS: Damage Severity

Class labels	Train	Dev	Test	Total
Disaster Type				
Earthquake	1,987	218	464	2,669
Fire	1,115	154	402	1,671
Flood	2,175	300	726	3,201
Hurricane	1,249	216	506	1,971
Landslide	917	127	287	1,331
Not disaster	3,064	564	1,463	5,091
Other disaster	489	218	870	1,577
Total	10,996	1,797	4,718	17,511
Informativeness				
Informative	22,018	2,736	6,578	31,332
Not informative	18,841	2,460	7,084	28,385
Total	40,859	5,196	13,662	59,717
Humanitarian				
Affected injured or dead people	537	115	353	1,005
Infrastructure and utility damage	2,397	736	2,095	5,228
Not humanitarian	4,354	886	2,964	8,204
Rescue volunteering or donation effort	1,312	268	752	2,332
Total	8,600	2,005	6,164	16,769
Damage Severity				
Little or none	9,124	1,677	4,149	14,950
Mild	3,188	663	1,339	5,190
Severe	11,102	1,145	2,509	14,756
Total	23,414	3,485	7,997	34,896

F. Multitask Learning

Since the tasks share similar properties, we also consider training the model in multitask settings with shared parameters. The benefits of multitask settings can be twofold: *(i)* learning shared representation can help the model generalize better and improve performance on individual tasks, and *(ii)* training a single model instead of four different models will yield a significant speed and reduce computational load during training and inference. It is important to mention that the *Crisis Benchmark Dataset* was not designed for multitask learning; rather, it was prepared for each task separately. Hence, we needed to prepare them for the multitask setup. Creating multitask learning datasets from *Crisis Benchmark Dataset*

TABLE VIII: Data split for multitask setting with **complete aligned labels** for the different combinations of two-tasks.

Class labels	Train	Dev	Test	Total
Informativeness & Humanitarian				
Informativeness				
Informative	2,111	399	1,064	3,574
Not informative	2,546	397	1,443	4,386
Total	4,657	796	2,507	7,960
Humanitarian				
Affected injured or dead people	426	72	166	664
Infrastructure and utility damage	410	81	210	701
Rescue volunteering or donation effort	1,274	246	688	2,208
Not humanitarian	2,547	397	1,443	4,387
Total	4,657	796	2,507	7,960
Informativeness & Damage Severity				
Informativeness				
Informative	14,683	1,306	2,206	18,195
Not informative	4,687	928	2,020	7,635
Total	19,370	2,234	4,226	25,830
Damage Severity				
Little or none	7,085	1,094	2,369	10,548
Mild	2,665	426	679	3,770
Severe	9,620	714	1,178	11,512
Total	19,370	2,234	4,226	25,830

TABLE IX: Data split for multi-task setting with **complete aligned labels** for four-tasks: Damage Types, Informativeness, Humanitarian, and Damage Severity.

Class labels	Train	Dev	Test	Total
Disaster Type				
Earthquake	68	25	90	183
Fire	80	35	155	270
Flood	102	54	162	318
Hurricane	110	75	214	399
Landslide	8	6	24	38
Other disaster	372	198	806	1,376
Not disaster	1,563	368	1,043	2,974
Total	2,303	761	2,494	5,558
Informativeness				
Informative	740	393	1,454	2,587
Not informative	1,563	368	1,040	2,971
Total	2,303	761	2,494	5,558
Humanitarian				
Affected injured or dead people	85	34	164	283
Infrastructure and utility damage	398	230	764	1,392
Rescue volunteering or donation effort	26	14	53	93
Not humanitarian	1,794	483	1,513	3,790
Total	2,303	761	2,494	5,558
Damage Severity				
Little or none	1,805	494	1,571	3,870
Mild	174	102	337	613
Severe	324	165	586	1075
Total	2,303	761	2,494	5,558

introduced a challenge – there is an overlap between train and test set images among different tasks. Hence, we prepare the datasets for the multitask setting using the following strategy:

1) We merge the test sets from different tasks into a combined test set. If an image in the combined test set is present in the train or dev set of some tasks, we remove it from that split and add the label of the task in the test set.
2) We merge the dev sets of the four tasks into the combined dev set. If an image in the combined dev set is present in the train set of some tasks, we remove it from that train split and add the label of the task in the dev set.
3) We merge the train sets of the four tasks into the combined train set. Since we have removed images that overlap with the dev set and test set in the previous steps, therefore, it guarantees that no image from the train set will be present in the other splits.

Since all the images do not have annotation for all four tasks, there is a discrepancy in the number of images available for different tasks. We report the distribution of the data splits for the multi-task setting in Table VII. Overall, there are 49353 images in the train set, 6157 images in the dev set, and 15688 images in the test set. Due to the overlap of images in different splits for different tasks, there is also a discrepancy between the number of images available between multi-task and single-task settings. As an example, for the disaster types task, there are 12846 images in the train set, 1470 images in the dev set, and 3195 images in the test set in the single-task setting. However, in the multi-task setting, these numbers are respectively 10996, 1797, and 4718. As a consequence of our merging procedure, there are more images in the test and dev sets and fewer images in the train set.

Few approaches have been proposed in the literature to address the issue of incomplete/missing labels in multi-task settings. They usually work by generating missing task labels using different methods, including Bayesian networks [82], rule-based approach [83], knowledge distillation from another model [84]. In our experiments, we opt for a simpler alternative. Specifically, we do not compute loss for a task if its label is missing. Since the tasks have varying training images, we calculate the loss for each task and aggregate them in a batch. This ensures that the loss of each task is weighted equally. The steps are detailed in Algorithm 1.

We also experiment with images having complete aligned labels for different tasks. We identified three such combinations that have a substantial number of images in different classes. Two of them belong to two task subsets. The first one is informativeness and humanitarian, which has 7,960 total aligned images. The second one is informativeness and damage severity, having 25,830 total images. Data distribution for these two settings is reported in Table VIII. The final subset of images having labels for all four tasks, which consists of 5558 images. Data distribution for this set is reported in Table IX.

V. Results

Our experimental results consist of different settings. Below we discuss each of them in detail.

Algorithm 1: Batch loss calculation in the multi-task setting

```
Input: batch_input    // images in the batch
       batch_labels   // list of labels for each task
       num_classes    // number of classes for each task
       model          // outputs prediction for all tasks are combined
Output: batch_loss
num_tasks = len(num_classes)
prediction = model.predict(batch_input)
batch_loss = 0
task_index = 0        // starting index for output corresponding to this task
for i ← 0 to num_tasks do
    prediction_task = prediction[:,
     task_index:task_index + num_classes[i]]
    label_task = batch_labels[i]
    /* if there is no label for a task
       it is marked as -1 in the label
       */
    valid_idx = nonzero(label_task != -1)
    task_loss =
     cross_entropy_loss(prediction_task[valid_idx],
     label_task[valid_idx])
    batch_loss = batch_loss + task_loss
    task_index = task_index + num_classes[i]
```

A. Dataset Comparisons

In Table X, we report classification results for different tasks and different datasets using ResNet18 network architecture. The performance of different tasks are not equally comparable as they have different levels of complexity (e.g., varying number of class labels, class imbalance, etc.). For example, the informativeness classification is a binary task, which is computationally simpler than a classification task with more labels (e.g., seven labels in disaster type). Hence, the performance is comparatively higher for informativeness. An example of a class imbalance issue can be seen in Table VI with the damage severity task. The distribution of mild is relatively small, which reflects on its and overall performance. The mild class label is also less distinctive than other class labels, and we noticed that classifiers often confuse this class label with the other two class labels. Similar findings have also been reported in [4]. For the disaster type task, the performance of the AIDR-DT model is higher compared to the DMD model. We observe that the DMD dataset is comparatively small, and the model is not performing well on the consolidated dataset. This characteristic is observed in other tasks as well. For the damage severity task, CrisisMMD is performing worse, which is also reflected in its dataset size, i.e., 2,493 images in the training set, as shown in Table V. As expected, overall, for all tasks, the models with the consolidated datasets outperform individual datasets.

TABLE X: Results on different classification tasks using the ResNet18 model. Trained on individual and consolidated datasets and tested on consolidated test sets.

Dataset	Acc	P	R	F1
Disaster Type (7 classes)				
AIDR-DT	0.76	0.72	0.76	0.73
DMD	0.58	0.73	0.58	0.59
Consolidated	0.79	0.78	0.79	**0.79**
Informativeness (2 classes)				
DAD	0.80	0.80	0.80	0.80
CrisisMMD	0.79	0.79	0.79	0.79
DMD	0.80	0.80	0.80	0.80
AIDR-Info	0.75	0.79	0.75	0.73
Consolidated	0.85	0.85	0.85	**0.85**
Humanitarian (4 classes)				
CrisisMMD	0.73	0.73	0.73	0.73
DMD	0.68	0.68	0.68	0.64
Consolidated	0.75	0.75	0.75	**0.75**
Damage Severity (3 classes)				
DAD	0.72	0.70	0.72	0.71
CrisisMMD	0.41	0.57	0.41	0.37
DMD	0.68	0.66	0.68	0.66
Consolidated	0.75	0.73	0.75	**0.74**

TABLE XI: Results using different neural network models on the consolidated dataset with four different tasks. Trained and tested using the consolidated dataset. Comparable results are shown in **bold** and best results are shown in underlined.

Architecture	Acc	P	R	F1	Acc	P	R	F1
	Disaster Type				**Informative**			
ResNet18	0.790	0.783	0.790	0.785	0.852	0.851	0.852	0.851
ResNet50	0.810	0.806	0.810	**0.808**	0.852	0.852	0.852	0.852
ResNet101	0.817	0.812	0.817	**0.813**	0.853	0.853	0.853	0.852
AlexNet	0.756	0.756	0.756	0.754	0.827	0.829	0.827	0.828
VGG16	0.800	0.796	0.800	0.798	0.859	0.858	0.859	**0.858**
DenseNet(121)	0.811	0.805	0.811	**0.806**	0.863	0.863	0.863	**0.862**
SqueezeNet	0.757	0.754	0.757	0.755	0.829	0.829	0.829	0.829
InceptionNet (v3)	0.562	0.609	0.562	0.528	0.663	0.723	0.663	0.593
MobileNet (v2)	0.785	0.781	0.785	0.782	0.850	0.849	0.850	0.849
EfficientNet (b1)	0.818	0.815	0.818	**0.816**	0.864	0.863	0.864	**0.863**
	Humanitarian				**Damage Severity**			
ResNet18	0.754	0.747	0.754	0.749	0.751	0.734	0.751	0.736
ResNet50	0.770	0.762	0.770	0.762	0.763	0.746	0.763	**0.751**
ResNet101	0.769	0.763	0.769	**0.765**	0.760	0.736	0.760	0.737
AlexNet	0.721	0.715	0.721	0.716	0.734	0.714	0.734	0.709
VGG16	0.778	0.773	0.778	**0.773**	0.769	0.750	0.769	**0.753**
DenseNet(121)	0.765	0.756	0.765	0.755	0.755	0.734	0.755	0.739
SqueezeNet	0.730	0.717	0.730	0.719	0.733	0.707	0.733	0.708
InceptionNet (v3)	0.598	0.637	0.598	0.509	0.660	0.623	0.660	0.615
MobileNet (v2)	0.751	0.745	0.751	0.746	0.746	0.727	0.746	0.730
EfficientNet (b1)	0.767	0.764	0.767	**0.765**	0.766	0.754	0.766	**0.758**

B. Network Architecture Comparisons

In Table XI, we report results using different network architectures on consolidated datasets for different tasks, i.e., trained and tested using a consolidated dataset. Across different tasks, EfficientNet (b1) is performing better than other models as shown in Figure 3, except for humanitarian task, for which VGG16 is outperforming other models. Comparatively the second-best models are VGG16, ResNet50, ResNet101, and DenseNet (101). From the results of different tasks, we observe that InceptionNet (v3) is the worst-performing model.

The performance difference among different models such

TABLE XII: Different neural network models with number of layer, parameters and memory requirement during the inference of a binary (Informativeness) classification task.

Model	# Layer	# Param (M)	Memory (MB)
ResNet18	18	11.18	74.61
ResNet50	50	23.51	233.54
ResNet101	101	42.50	377.58
AlexNet	8	57.01	222.24
VGG16	16	134.28	673.87
DenseNet (121)	121	6.96	174.2
SqueezeNet	18	0.74	47.99
InceptionNet (v3)	42	24.35	206.01
MobileNet (v2)	20	2.23	8.49
EfficientNet (b1)	25	7.79	177.82

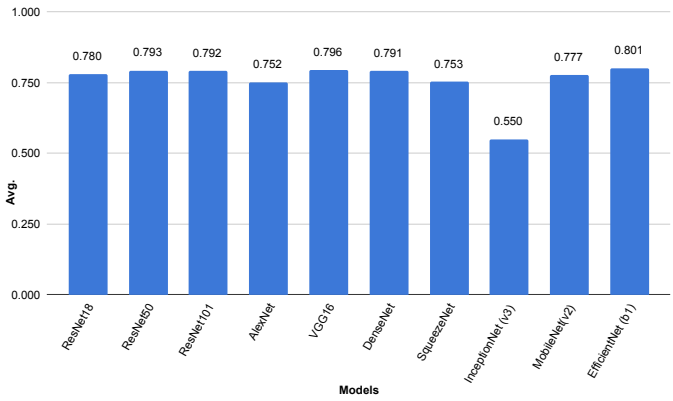


Fig. 3: Average F1 scores from all four tasks with different network architectures show that on average EfficientNet (b1) performs better than other architectures.

as EfficientNet (b1), VGG16, ResNet50, ResNet101, and DenseNet (101) are low, hence, we have done statistical test to understand whether such small differences are significant. We used McNemar's test for binary classification task, (i.e., informativeness) and Bowker's test for other multiclass classification tasks. More details of this test can be found in [85]. We have done such tests between two models to see a pairwise difference. In Figure 4, we report the results of significant tests. The value in the cell represent the P-value and the light yellow color represent they are statistically significant with $P < 0.05$. From the Figure 4, we see that for disaster type task the P-value is higher than 0.05 in comparison between EfficientNet (b1) vs. ResNet50, ResNet101 and DenseNet (121), which clearly reflects among the results reported in Table XI. Similarly the difference is very low between EfficientNet (b1) vs. VGG16 and DenseNet (121). For humanitarian and damage severity tasks, we observed similar behaviors. By analyzing all four tasks it appears VGG16 is the second best performing model.

In Table XII, we also report different neural network models with their number of layers, parameters, and memory consumption during the inference of informativeness task. There is usually a trade-off between the performance and computational complexity of different deep neural networks. In terms of memory consumption and the number of parameters, VGG16 is more expensive than others. Among different ResNet models, ResNet18 is a reasonable choice, given that its computational complexity is significantly less than other ResNet models. Based on the performance and computational complexity, we can conclude that EfficientNet can be the best option for real-time applications. We computed throughput for EfficientNet on a Tesla T4 GPU using a batch size of 16, and it can process $\sim$191 images per second in a single task setting as opposed to $\sim$743 in a multitask setting. We also computed the same on the CPU with a batch size of 1 in a single thread. It can process $\sim$1.6 and $\sim$6 images in a single task and multitask setting, respectively.

C. Data Augmentation

To reduce the overfitting and to have more generalized models, we used data augmentation and weight decay. In Table XIII, we report the results for all tasks and using all network architectures. The column ***Diff.*** reports the difference between the results presented in Table XI where no RandAugment or *weight decay* has been applied. The improved results are highlighted with light blue color for all tasks. Out of 40 experiments (10 network architectures across 4 tasks), for 26 cases, the augmentation with weight decay improved the performances.

On the improved cases, we also computed a statistical significance test between no RandAugment and RandAugment with *weight decay* models. We found that the improvements for the models with InceptionNet (v3) are statistically significant in all tasks. Only the improved performance with EfficientNet (b1) for damage severity task is statistically significant, and for other tasks, they are not statistically significant. We investigated training and validation losses over the number of epochs. In Figure 5 and 6, we report training, validation losses and accuracies for EfficientNet (b1) model for Informativeness and Humanitarian tasks, respectively. From the figures 5a and 6a, we clearly see that models are overfitting, whereas Figures 5b and 6b show that models are more generalized. These findings demonstrate the benefits of augmentation and weight decay.

D. Semi-supervised Learning

In Table XIV, we present the results of the Noisy student-based self-training approach without/with RandAugment results. We have an $\sim 1\%$ improvement for the *Informativeness* task. For the *Humanitarian* task, the performance is similar to RandAugment. For the *Damage severity* task, the performance of Noisy student is the same as without RandAugment but lower than RandAugment.

We postulate the following possible reasons for the lack of improvements in semi-supervised learning experiments:

1) Semi-supervised learning usually performs better when trained from scratch instead of fine-tuning from a pre-trained model. This phenomenon is explored in [86] where the authors reported the performance gained from semi-supervised learning methods are usually smaller when trained from a pretrained model. We could not train the student model from scratch as our labeled datasets are small, and it degrades performance even more.

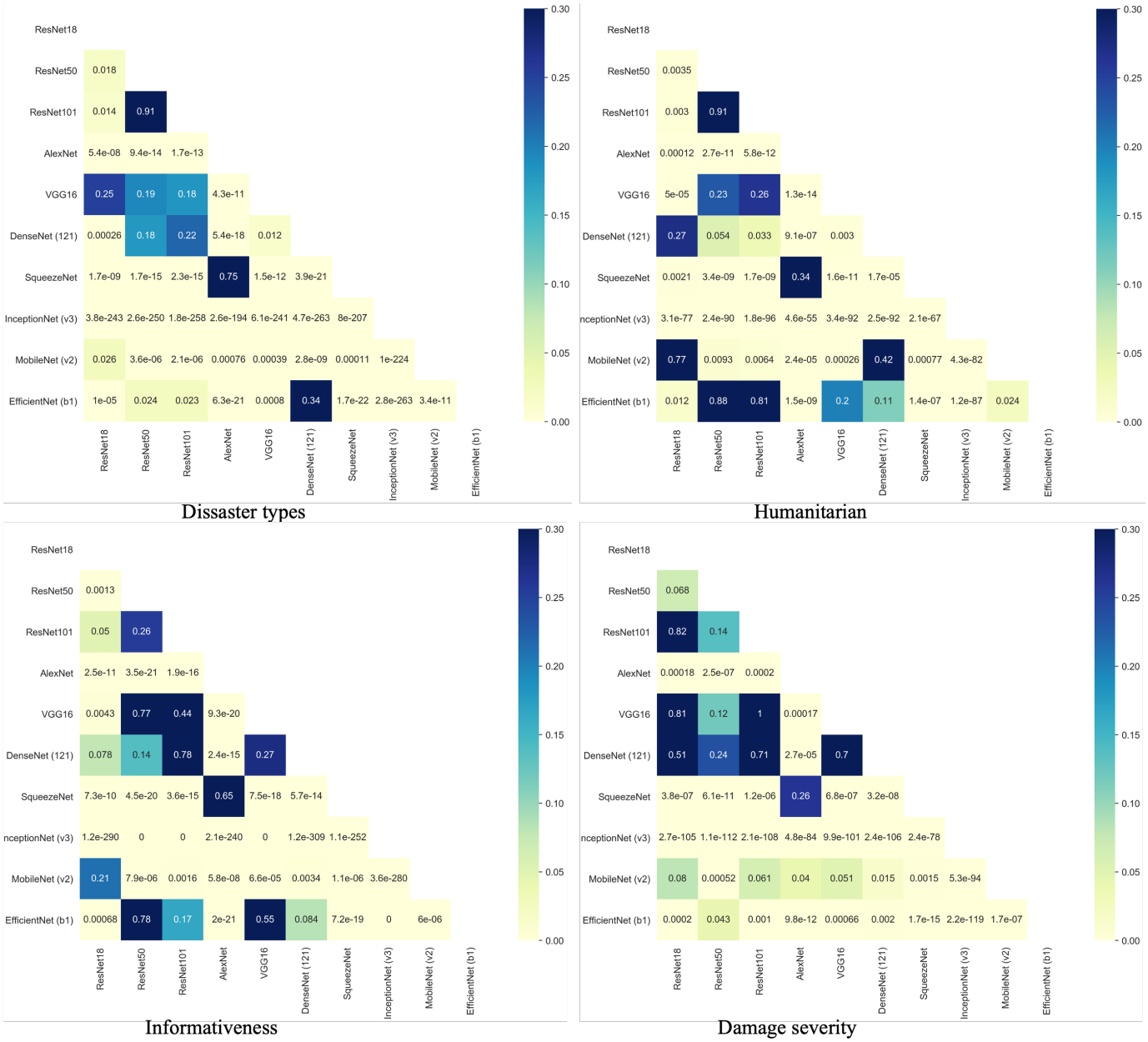


Fig. 4: Statistical significant test among the different network architectures for *Disaster Type*, *Informativeness*, *Humanitarian* and *Damage Severity* tasks. P-values are presented in cells. Light yellow color represent they are statistically significant with $p < 0.05$

2) We had to use a much smaller labeled batch size of 16 compared to those used in [69] (512 or higher) due to GPU constraints. Having a larger labeled batch size and, consequently, more unlabeled images in each batch may yield a better result.

E. Multitask Learning

Since the *Crisis Benchmark Dataset* has not been designed to address the multitask learning, we needed to re-split it as discussed in Section IV-F. This resulted two different settings: *(i)* incomplete/missing labels, and *(ii)* complete aligned labels. The incomplete/missing labels in multitask learning is a challenging problem, which we addressed using masking, i.e., for an unlabeled output, we are not computing loss for that particular task. In Table XV, we report the results of multitask learning with missing labels where we address all tasks. We also investigated different task combinations where all labels are present. In Table XVI, we report the results of different tasks combinations where they have complete aligned labels. For different task combinations, performances differ due to their data sizes, label distribution, and task settings. The results with multitask learning are not directly comparable with our single task setup. However, they can serve as a baseline for future studies.

TABLE XIII: Results with data augmentation and weight decay using different neural network models on the consolidated dataset for all four tasks. **Diff.** represents the difference F1 score without RandAugment results presented in Table XI. * represents statistically significant (with $P < 0.05$) compared to the without RandAugment results.

Architecture	Acc	P	R	F1	Diff.	Acc	P	R	F1	Diff.
	Disaster Type					**Informative**				
ResNet18	0.812	0.807	0.812	0.809	0.024	0.848	0.847	0.848	0.847	-0.004
ResNet50	0.817	0.81	0.817	0.812	0.004	0.863	0.863	0.863	0.862	0.010
ResNet101	0.819	0.815	0.819	0.816	0.003	0.857	0.858	0.857	0.858	0.006
AlexNet	0.755	0.753	0.755	0.753	-0.001	0.827	0.826	0.827	0.825	-0.003
VGG16	0.803	0.797	0.803	0.798	0.000	0.855	0.855	0.855	0.855	-0.003
DenseNet (121)	0.817	0.811	0.817	0.813	0.007	0.858	0.858	0.858	0.857	-0.005
SqueezeNet	0.726	0.719	0.726	0.717	-0.038	0.821	0.820	0.821	0.820	-0.009
InceptionNet (v3)	0.808	0.801	0.808	*0.802	0.254	0.860	0.859	0.860	*0.859	0.331
MobileNet (v2)	0.793	0.788	0.793	0.789	0.007	0.854	0.853	0.854	0.853	0.004
EfficientNet (b1)	0.838	0.834	0.838	**0.835**	0.019	0.869	0.868	0.869	**0.868**	0.005
	Humanitarian					**Damage Severity**				
ResNet18	0.745	0.738	0.745	0.741	-0.008	0.757	0.736	0.757	0.739	0.003
ResNet50	0.774	0.769	0.774	0.768	0.006	0.763	0.745	0.763	0.749	-0.002
ResNet101	0.774	0.778	0.774	0.775	0.010.	0.766	0.753	0.766	0.757	0.020
AlexNet	0.718	0.709	0.718	0.709	-0.007	0.728	0.712	0.728	0.713	0.004
VGG16	0.772	0.766	0.772	0.767	-0.006	0.767	0.748	0.767	0.752	-0.001
DenseNet (121)	0.759	0.756	0.759	0.755	0.000	0.760	0.741	0.760	0.747	0.008
SqueezeNet	0.720	0.713	0.720	0.712	-0.007	0.729	0.708	0.729	0.702	-0.006
InceptionNet (v3)	0.762	0.753	0.762	*0.754	0.256	0.758	0.735	0.758	*0.739	0.115
MobileNet (v2)	0.759	0.749	0.759	0.751	0.005	0.758	0.737	0.758	0.738	0.008
EfficientNet (b1)	0.785	0.784	0.785	**0.784**	0.019	0.777	0.762	0.777	***0.765**	0.007

TABLE XIV: Results with Noisy student self-training approach using Efficient (b1) neural network models on the consolidated datasets for all four tasks.

Experiment	Acc	P	R	F1
	Disaster Type			
Without RandAugment	0.818	0.815	0.818	0.816
RandAugment	0.838	0.834	0.838	0.835
Noisy Student	0.793	0.812	0.793	0.794
	Informativeness			
Without RandAugment	0.864	0.863	0.864	0.863
RandAugment	0.869	0.868	0.869	0.868
Noisy Student	0.878	0.878	0.878	**0.876**
	Humanitarian			
Without RandAugment	0.767	0.764	0.767	0.765
RandAugment	0.785	0.784	0.785	0.784
Noisy Student	0.783	0.786	0.783	0.783
	Damage Severity			
Without RandAugment	0.766	0.754	0.766	0.758
RandAugment	0.777	0.762	0.777	0.765
Noisy Student	0.773	0.753	0.773	0.759

TABLE XV: Results of multitask learning with **incomplete/missing** labels.

Task	Acc	P	R	F1
Disaster type	0.647	0.657	0.647	0.637
Informativeness	0.727	0.735	0.727	0.726
Humanitarian	0.775	0.772	0.775	0.773
Damage severity	0.744	0.732	0.744	0.737

F. Visual Explanation using Grad-CAM

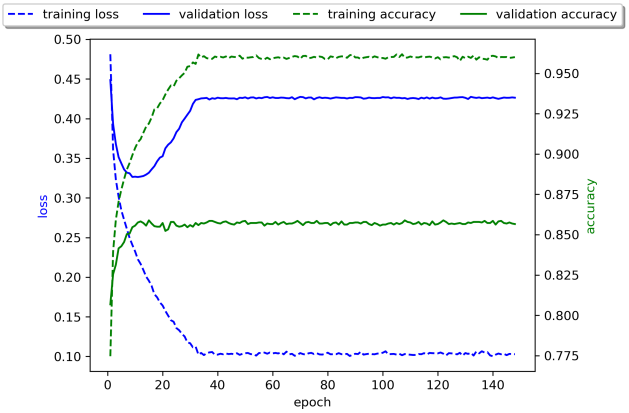
We explore how the neural networks arrive at their decision by utilizing Gradient-weighted Class Activation Mapping (Grad-CAM) [9]. Grad-CAM uses the gradient of a target class flowing into the final convolution layer to produce a localization map highlighting the important regions in the image for that specific class. We report results for two candidate networks, i.e., VGG16 and EfficientNet, on two tasks, i.e., informativeness and disaster type. We use the models trained using RandAugment for this experiment.

In Figure 7, we show the activation map for the predicted class for some images from the informativeness test set. From these images, it is apparent that EfficientNet performs better for localizing important regions in the image for the class of interest. VGG16 tends to depend on smaller regions for decision-making. The last row shows an image where VGG16 misclassified an informative image as not informative.

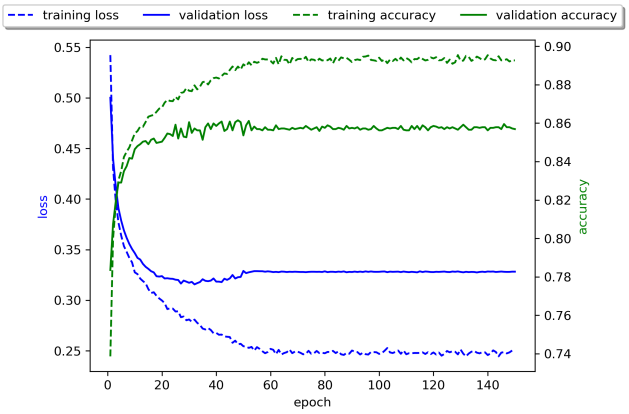
We show the activation map for some images from the test set of the disaster type task in Figure 8. Here, the difference

TABLE XVI: Results of multitask learning with different tasks combinations and **complete labels**. DT: Disaster Type, Info: Informative, Hum: Humanitarian, DS: Damage Severity.

Task	Acc	P	R	F1
Two tasks: Info and DS				
Informative	0.855	0.856	0.855	0.855
Damage Severity	0.806	0.799	0.806	0.802
Two tasks: Info and Hum				
Informative	0.817	0.816	0.817	0.816
Humanitarian	0.761	0.756	0.761	0.758
Four tasks: DT, Info, Hum and DS				
Disaster Type	0.781	0.768	0.781	0.772
Informative	0.920	0.921	0.920	0.920
Humanitarian	0.827	0.807	0.827	0.816
Damage Severity	0.772	0.750	0.772	0.759

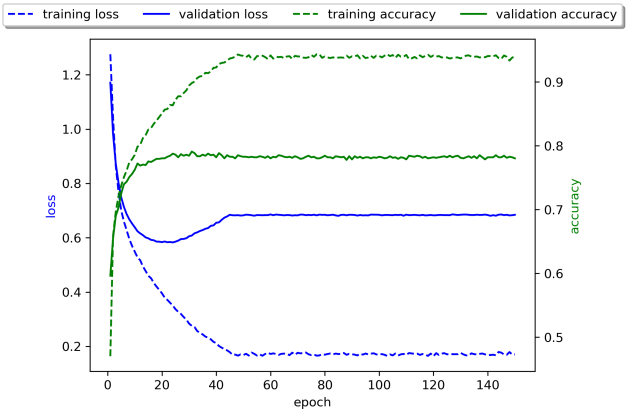


(a) Without RandAugment.

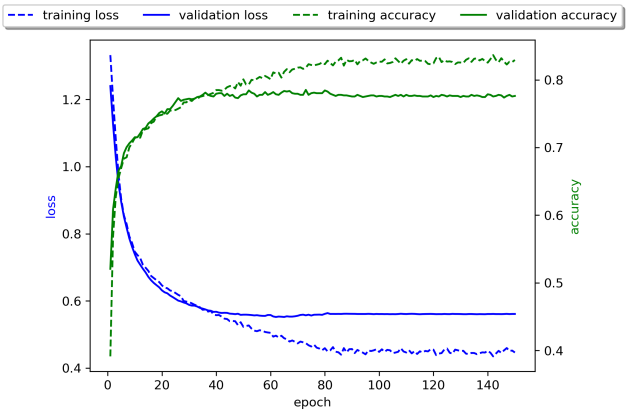


(b) With RandAugment and weight decay.

Fig. 5: Training/validation losses and accuracies without and with augmentation for ***Informativeness*** task.



(a) Without RandAugment.



(b) With RandAugment and weight decay.

Fig. 6: Training/validation losses and accuracies without and with augmentation for ***Humanitarian*** task.

in localization quality between the two models is even more pronounced. The activation maps from VGG are difficult to interpret in the first and third images, even though the model classifies them correctly. The second image shows that VGG may focus on the smoke regions for classifying fire images. This explains why it identifies the last image as fire, misclassifying the clouds as smoke.

Overall, these results suggest that EfficientNet does not only outperform other models in the numeric measures but it also produces activation maps that are easier to interpret.

VI. Discussion and Future Work

A. Our Findings

Real-time event detection is an important problem from social media content. Our proposed pipeline and models are suitable to deploy them in different applications. The proposed models can also be used independently. For example, disaster type model can be used to monitor disaster events.

Our experiments were based on the research questions discussed in Section I below we report our findings based on them.

RQ1: Our investigation to dataset comparison suggests that data consolidation helps, which answers our first research question.

RQ2: We also explore several deep learning models, which vary with performance and complexities. Among them, EfficientNet (b1) appears to be a reasonable option. Note that EfficientNet has a series of network architectures (b0-b7) and for this study, we only reported results with EfficientNet (b1). We aim to further explore other architectures. A small and low latency model is desired to deploy mobile and handheld embedded computer vision applications. The development of MobileNet [66] sheds light towards that direction. Our experimental results suggest that it is computationally simpler and provides a reasonable accuracy, only 2-3% lower than the best models for different tasks. These findings answer out second research question.

RQ3: We observe that strong data augmentation can improve performance, although this is not consistent across different tasks and models. Semi-supervised learning does not usually yield performance when trained using pretrained models and can sometimes even degrade it.

RQ4: Multi-task learning can be an ideal solution for the real-time system as it can potentially provide speed-ups of

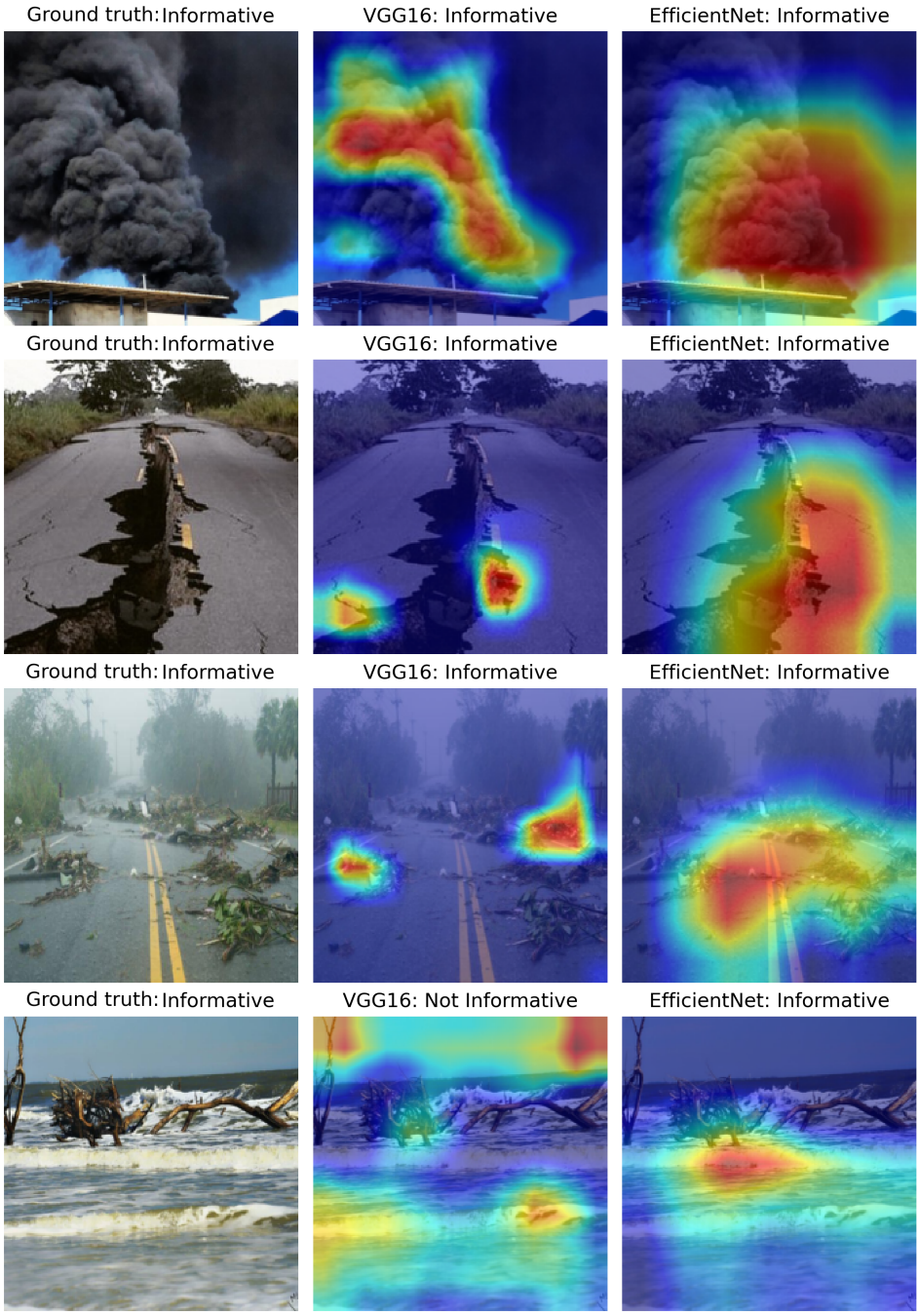


Fig. 7: Grad-CAM visualization of some images for the informativeness task.

multiple factors during inference. However, some tasks may perform worse than their single task settings in the presence of incomplete labels. Having aligned complete labels for different tasks can mitigate this issue.

B. Comparison with the State of the Art

We compared our results with recent and related state-of-the-art results, reported in Table XVII. However, it is not possible to have an end-to-end comparison for a few possible reasons: *(i)* different datasets and sizes – see the second and third columns in Table XVII, *(ii)* different data splits (train/dev/test *vs.* Cross Validation (CV) fold) even using same dataset – see the *Data Split* column in the same Table, *(iii)* different evaluation measures such as weighted P/R/F1-measure (first two rows) [39] *vs.* accuracy (third row) [5] *vs.* CV fold (fourth to sixth rows – unspecified in [31] whether measures are macro, micro or weighted).

Even if they are not exactly comparable, we observe that on informativeness and humanitarian tasks, previously reported results (weighted F1) are 0.832 and 0.763, respectively, using the CrisisMMD dataset [39]. The authors in [5] reported a test accuracy of 0.840 ± 0.0172 for six disaster types tasks using the

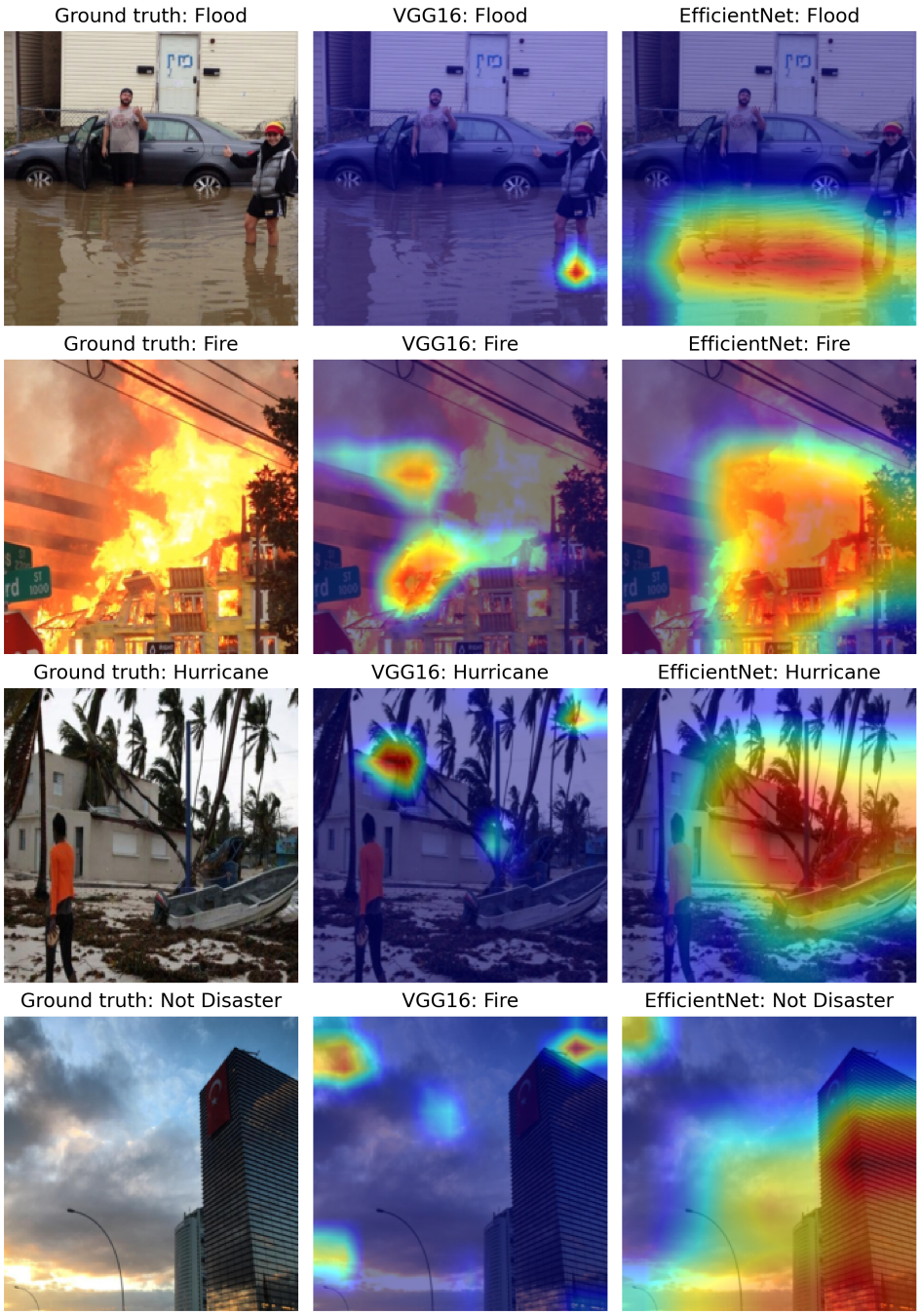


Fig. 8: Grad-CAM visualization of some images for the disaster type task.

DMD dataset with a five-fold cross-validation run. The study in [31] report an F1 of 0.820 for informativeness, 0.920 for infrastructure damage, and 0.940 for damage severity. In another study, using the CrisisMMD dataset, authors report weighted-F1 of 0.812 and 0.870 for informativeness and humanitarian tasks, respectively [32]. They used a small subset of the whole CrisisMMD dataset in their study. From the Table XVII we observe that the F1 for informativeness task ranges from 0.812 to 0.832 across studies, for humanitarian task it varies from 0.763 to 0.870, and for damage severity it varies from 0.661 to 0.940. Compared to them our best results (weighted F1) for disaster types, informativeness, humanitarian and damage severity are 0.835, 0.876, 0.784, and 0.765, respectively, on the consolidated single task dataset.

C. Future Work

As for future work we foresee several interesting research avenues. *(i)* Further exploration of semi-supervised learning to leverage a large amount of unlabeled social media data and address the limitations highlighted in Section V-D. We believe addressing such limitations can help to advance state of the art. *(ii)* In multitask setup, one possible research direction

TABLE XVII: **Recent relevant results reported in the literature**. *# C*: Number of class labels, Cls: Classification task, B: Binary, M: Multiclass, Info: Informativeness, Hum: Humanitarian, Event: Disaster event types, Infra.: Infrastructural damage, Severity: Severity Assessment. We converted some numbers from percentage (as originally reported) to decimal for an easier comparison.

Ref.	Dataset	# image	# C	Cls.	Task	Models	Data Split	Acc	P	R	F1
[39]	CrisisMMD	12,708	2	B	Info	VGG16	Train/dev/test	0.833	0.831	0.833	0.832
[39]	CrisisMMD	8,079	5	M	Hum	VGG16	Train/dev/test	0.768	0.764	0.768	0.763
[5]	DMD	5879	6	M	Event	InceptionNet (v4)	4 folds CV	0.840	-	-	-
[31]	CrisisMMD	18,126	2	B	Info	InceptionNet (v4)	5 folds CV	-	0.820	0.820	0.820
[31]	CrisisMMD	18,126	2	B	Infra.	InceptionNet (v4)	5 folds CV	-	0.920	0.920	0.920
[31]	CrisisMMD	18,126	3	B	Severity	InceptionNet (v4)	5 folds CV	-	0.950	0.940	0.940
[32]	CrisisMMD	11,250	2	B	Info	DenseNet	Train/dev/test	0.816	-	-	0.812
[32]	CrisisMMD	3,359	5	B	Hum	DenseNet	Train/dev/test	0.834	-	-	0.870
[32]	CrisisMMD	3,288	3	B	Severity	DenseNet	Train/dev/test	0.629	-	-	0.661

is to address the problem of incomplete/missing labels, and the other is manually labeling *Crisis Benchmark Dataset* for incomplete labels for all tasks. Both approaches will give the community grounds to explore multitask learning for real-time social media image classification.

VII. Applications

There are many application scenarios of the proposed models, however, in this section we discuss the ones that are highly relevant for crisis responders in humanitarian organizations.

Information for Situational Awareness: The information posted on social media during natural or human-induced disasters varies greatly. Studies have revealed that a big proportion of social media data consists of irrelevant information that is not useful for any kind of relief operations. For the decision-making process, humanitarian organizations are interested to have concise information about the ongoing situation to be aware of the event. The proposed models can help in filtering and reducing irrelevant content and provide a concrete summary.

Actionable Information: Depending on their roles and mandate, humanitarian organizations differ in terms of their information needs. Several rapid response and relief agencies look for fine-grained information about specific incidents, which is also actionable. Such information types include reports of injured or dead people, critical infrastructure damage (e.g., a collapsed bridge), and rescue demand among others. Our study focused on coarse (i.e., binary) to fine-grained labels while also addressed four different but related tasks. Applications can be developed on top of our models, which can provide critical humanitarian information needs in crisis situations.

Real-time Crisis Event Detection: The proposed models (i.e., disaster type) can be deployed to continuously monitor social media and detect emergent events (e.g., fire, flood) around the world.

VIII. Conclusions

The imagery and textual content available on social media have been used by humanitarian organizations in times of disaster events. There has been limited work for disaster response image classification tasks compared to text. In this study, we posed four research questions and performed extensive experiments on four tasks such as disaster type, informativeness, humanitarian, and damage severity to answer those questions. Our experimental results on individual and consolidated datasets suggest that data consolidation helps. We investigated four tasks using various state-of-the-art neural network architectures and reported the best-performing models. The findings on data augmentation suggest that a more generalized model can be obtained with such approaches. Our investigation on semi-supervised and multitask learning suggests new research directions for the community. We also provide some insights of activation maps to demonstrate what class-specific information is learned by the network.

Funding

Open access funding information will be available upon approval.

Compliance with ethical standards

a) Conflict of interest: We have no conflicts of interest or competing interests to declare.

b) Availability of data and material: The data used in this study are available at https://crisisnlp.qcri.org/crisis-image-datasets-asonam20.

Data Availability

The dataset proposed in this research is available to download from the following links: https://crisisnlp.qcri.org/crisis-image-datasets-asonam20 and https://doi.org/10.7910/DVN/QXT5QL. We aim to maintain the data for a long period of time and make sure dataset is accessible.

References

[1] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: A survey," *ACM Computing Surveys*, vol. 47, no. 4, p. 67, 2015.

[2] N. Said, K. Ahmad, M. Riegler, K. Pogorelov, L. Hassan, N. Ahmad, and N. Conci, "Natural disasters detection in social media and satellite imagery: a survey," *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 31 267–31 302, 2019.

[3] 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," *Information Processing & Management*, vol. 57, no. 5, p. 102261, 2020.

[4] D. T. Nguyen, F. Ofli, M. Imran, and P. Mitra, "Damage assessment from social media imagery data during disasters," in *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Aug 2017, pp. 1–8.

[5] H. Mouzannar, Y. Rizk, and M. Awad, "Damage Identification in Social Media Posts using Multimodal Deep Learning," in *Proceedings of the 15th ISCRAM Conference*, Rochester, NY, USA, May 2018, pp. 529–543.

[6] F. Alam, F. Ofli, M. Imran, T. Alam, and U. Qazi, "Deep learning benchmarks and datasets for social media image classification for disaster response," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2020, pp. 151–158.

[7] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

[8] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013.

[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[10] R. Peters and J. P. de Albuqerque, "Investigating images as indicators for relevant social media messages in disaster management," in *Proc. of ISCRAM*, 2015.

[11] S. Daly and J. Thom, "Mining and classifying image posts on social media to analyse fires," in *Proceedings of the ISCRAM 2016 Conference*, Rio de Janeiro, Brazil, 2016, pp. 1–14.

[12] D. T. Nguyen, F. Alam, F. Ofli, and M. Imran, "Automatic image filtering on social networks using deep learning and perceptual hashing during crises," in *Proceedings of the 14th ISCRAM Conference*, Albi, France, May 2017.

[13] F. Alam, M. Imran, and F. Ofli, "Image4Act: Online social media image processing for disaster response," in *2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2017, pp. 601–604.

[14] F. Alam, F. Ofli, and M. Imran, "Processing social media images by combining human and machine computing during crises," *International Journal of Human Computer Interaction*, vol. 34, no. 4, pp. 311–327, 2018.

[15] F. Alam, U. Qazi, M. Imran, and F. Ofli, "Humaid: Human-annotated disaster incidents data from twitter with deep learning benchmarks," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 15, 2021, pp. 933–942.

[16] F. Alam, H. Sajjad, M. Imran, and F. Ofli, "CrisisBench: Benchmarking crisis-related social media datasets for humanitarian information processing," in *Proceedings of the International AAAI Conference on Web and Social Media*, 2021.

[17] K. R. Nia and G. Mori, "Building damage assessment using deep learning and ground-level image data," in *14th Conference on Computer and Robot Vision (CRV)*. IEEE, 2017, pp. 95–102.

[18] X. Li, D. Caragea, H. Zhang, and M. Imran, "Localizing and quantifying damage in social media images," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018, pp. 194–201.

[19] M. Imran, F. Alam, U. Qazi, S. Peterson, and F. Ofli, "Rapid damage assessment using social media images by combining human and machine intelligence," in *Proceedings of the 17th ISCRAM Conference*, Blacksburg, VA, USA, 2020.

[20] T. Chen, D. Lu, M.-Y. Kan, and P. Cui, "Understanding and classifying image tweets," in *ACM Multimedia*, 2013, pp. 781–784.

[21] F. Alam, F. Ofli, and M. Imran, "Processing social media images by combining human and machine computing during crises," *International Journal of Human–Computer Interaction*, vol. 34, no. 4, pp. 311–327, 2018.

[22] X. Li, D. Caragea, C. Caragea, M. Imran, and F. Ofli, "Identifying disaster damage images using a domain adaptation approach," in *Proceedings of the 16th ISCRAM Conference*, València, Spain, 2019, pp. 633–645.

[23] S. Pouyanfar, Y. Tao, S. Sadiq, H. Tian, Y. Tu, T. Wang, S.-C. Chen, and M.-L. Shyu, "Unconstrained flood event detection using adversarial data augmentation," in *IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 155–159.

[24] S. Ahmad, K. Ahmad, N. Ahmad, and N. Conci, "Convolutional neural networks for disaster images retrieval." in *MediaEval*, 2017.

[25] R. Lagerstrom, Y. Arzhaeva, P. Szul, O. Obst, R. Power, B. Robinson, and T. Bednarz, "Image classification to support emergency situation awareness," *Frontiers in Robotics and AI*, vol. 3, p. 54, 2016.

[26] H. Ning, Z. Li, M. E. Hodgson *et al.*, "Prototyping a social media flooding photo screening system based on deep learning," *ISPRS International Journal of Geo-Information*, vol. 9, no. 2, p. 104, 2020.

[27] S. Z. Hassan, K. Ahmad, A. Al-Fuqaha, and N. Conci, "Sentiment analysis from images of natural disasters," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 104–113.

[28] K. Ahmad, M. Riegler, K. Pogorelov, N. Conci, P. Halvorsen, and F. De Natale, "Jord: a system for collecting information and monitoring natural disasters by linking social media with satellite imagery," in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, 2017, pp. 1–6.

[29] R. I. Jony, A. Woodley, and D. Perrin, "Flood detection in social media images using visual features and metadata," *2019 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2019.

[30] P. Kumar, F. Ofli, M. Imran, and C. Castillo, "Detection of disaster-affected cultural heritage sites from social media images using deep learning techniques," *J. Comput. Cult. Herit.*, vol. 13, no. 3, Aug. 2020.

[31] M. Agarwal, M. Leekha, R. Sawhney, and R. R. Shah, "Crisis-DIAS: towards multimodal damage analysis - deployment, challenges and assessment," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 346–353, Apr. 2020.

[32] M. Abavisani, L. Wu, S. Hu, J. Tetreault, and A. Jaimes, "Multimodal categorization of crisis events in social media," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2020, pp. 14 667–14 677.

[33] X. Huang, C. Wang, Z. Li, and H. Ning, "A visual–textual fused approach to automated tagging of flood-related tweets during a flood event," *International Journal of Digital Earth*, vol. 12, no. 11, pp. 1248–1264, 2019.

[34] X. Huang, Z. Li, C. Wang, and H. Ning, "Identifying disaster related social media for rapid response: a visual-textual fused cnn architecture," *International Journal of Digital Earth*, 2019.

[35] Y. Feng and M. Sester, "Extraction of pluvial flood relevant volunteered geographic information (vgi) by deep learning from user generated texts and photos," *ISPRS International Journal of Geo-Information*, vol. 7, no. 2, p. 39, 2018.

[36] T. Chen and C. Guestrin, "XGboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[37] Y. Rizk, H. S. Jomaa, M. Awad, and C. Castillo, "A computationally efficient multi-modal classification approach of disaster-related twitter images," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2050–2059.

[38] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3485–3492.

[39] F. Ofli, F. Alam, and M. Imran, "Analysis of social media data using multimodal deep learning for disaster response," in *Proceedings of the 16th ISCRAM Conference*, May 2020.

[40] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

[41] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proc. of CVPR Workshops*, 2014, pp. 806–813.

[42] G. Ozbulak, Y. Aytar, and H. K. Ekenel, "How transferable are cnn-based features for age and gender classification?" in *International Conference of the Biometrics Special Interest Group*, Sept 2016, pp. 1–6.

[43] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. of CVPR*, 2014, pp. 1717–1724.

[44] F. Alam, Z. Hassan, K. Ahmad, A. Gul, M. A. Riegler, N. Conci, and A. Al-Fuqaha, "Flood detection via twitter streams using textual and visual features," *Multimediaeval Benchmark 2020*, 2020.

[45] F. Alam, T. Alam, M. A. Hasan, A. Hasnat, M. Imran, and F. Ofli, "MEDIC: a multi-task learning dataset for disaster image classification," *Neural Computing and Applications*, Sep 2022.

[46] M. Imran, U. Qazi, F. Ofli, S. Peterson, and F. Alam, "Ai for disaster rapid damage assessment from microblogs," in *Thirty-Fourth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-22)*, 2022.

[47] S. Z. Hassan, K. Ahmad, S. Hicks, P. Halvorsen, A. Al-Fuqaha, N. Conci, and M. Riegler, "Visual sentiment analysis from disaster images in social media," *Sensors*, vol. 22, no. 10, 2022.

[48] C. Kyrkou and T. Theocharides, "Deep-learning-based aerial image classification for emergency response applications using unmanned aerial vehicles," in *CVPR Workshops*, 2019, pp. 517–525.

[49] Z. Zou, H. Gan, Q. Huang, T. Cai, and K. Cao, "Disaster image classification by fusing multimodal social media data," *ISPRS International Journal of Geo-Information*, vol. 10, no. 10, 2021.

[50] F. Alam, F. Ofli, and M. Imran, "CrisisMMD: multimodal twitter datasets from natural disasters," in *Twelfth international AAAI conference on web and social media*, Jun 2018, pp. 465–473.

[51] B. Bischke, P. Helber, C. Schulze, V. Srinivasan, A. Dengel, and D. Borth, "The multimedia satellite task at MediaEval 2017." in *In Proceedings of the MediaEval 2017: MediaEval Benchmark Workshop*, 2017.

[52] B. Benjamin, H. Patrick, Z. Zhengyu, B. J. de, and B. Damian, "The multimedia satellite task at MediaEval 2018: Emergency response for flooding events," in *MediaEval*, Oct 2018.

[53] R. Gupta, B. Goodman, N. Patel, R. Hosfelt, S. Sajeev, E. Heim, J. Doshi, K. Lucas, H. Choset, and M. Gaston, "Creating xbd: A dataset for assessing building damage from satellite imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[54] E. Weber, N. Marzo, D. P. Papadopoulos, A. Biswas, A. Lapedriza, F. Ofli, M. Imran, and A. Torralba, "Detecting natural disasters, damage, and incidents in the wild," in *European Conference on Computer Vision*. Springer, 2020, pp. 331–350.

[55] I. M. Shaluf, "Disaster types," *Disaster Prevention and Management: An International Journal*, 2007.

[56] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "AIDR: Artificial intelligence for disaster response," in *Proceedings of the 23rd international conference on world wide web*, 2014, pp. 159–162.

[57] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg, "Crisislex: A lexicon for collecting and filtering microblogged communications in crises." in *Proc. of ICWSM*, 2014.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.

[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[63] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[64] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size," *arXiv:1602.07360*, 2016.

[65] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[66] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, 2017.

[67] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv:1905.11946*, 2019.

[68] J. Moody, S. Hanson, A. Krogh, and J. A. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, no. 1995, pp. 950–957, 1995.

[69] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 687–10 698.

[70] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proceedings of the Advances in Neural Information Processing Systems 33 pre-proceedings (NeurIPS 2020)*, 2020.

[71] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "ReMixMatch: Semi-supervised learning with distribution matching and augmentation anchoring," *arXiv preprint arXiv:1911.09785*, 2019.

[72] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in neural information processing systems*, vol. 32, 2019.

[73] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[74] G. J. McLachlan, "Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis," *Journal of the American Statistical Association*, vol. 70, no. 350, pp. 365–369, 1975.

[75] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[76] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[77] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, A. Solin, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *Neural Networks*, vol. 145, pp. 90–106, 2022.

[78] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," *arXiv preprint arXiv:1904.12848*, 2019.

[79] F. Alam, S. Joty, and M. Imran, "Graph based semi-supervised learning with convolution neural networks to classify crisis related tweets," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, no. 1, 2018.

[80] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.

[81] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[82] A. Kapoor, R. Viswanathan, and P. Jain, "Multilabel classification using bayesian compressed sensing," *Advances in neural information processing systems*, vol. 25, pp. 2645–2653, 2012.

[83] D. Kollias and S. Zafeiriou, "Expression, affect, action unit recognition: Aff-wild2, multi-task learning and arcface," *arXiv preprint arXiv:1910.04855*, 2019.

[84] D. Deng, Z. Chen, and B. E. Shi, "Multitask emotion recognition with incomplete labels," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*. IEEE Computer Society, 2020, pp. 828–835.

[85] J. I. Hoffman, "Chapter 15 - categorical and cross-classified data: Mcnemar's and bowker's tests, kolmogorov-smirnov tests, concordance," in *Basic Biostatistics for Medical and Biomedical Practitioners (Second Edition)*, second edition ed., J. I. Hoffman, Ed. Academic Press, 2019, pp. 233 – 247.

[86] H.-Y. Zhou, A. Oliver, J. Wu, and Y. Zheng, "When semi-supervised learning meets transfer learning: Training strategies, models and datasets," *arXiv preprint arXiv:1812.05313*, 2018.

IX. Biography Section

Firoj Alam is a Scientist at the Qatar Computing Research Institute, HBKU. He received his PhD from the University of Trento, Italy, and has been working in Artificial Intelligence, Deep/Machine learning, Natural Language Processing, Social media content, Image Processing, and Conversation Analysis. His current research includes disinformation detection (http://tanbih.qcri.org), fact-checking, multimodal propaganda detection. He is an IEEE senior member with over 70 publications in refereed conferences and journals. He won the ISCRAM Best CoRe Paper Award in 2020, the ISCRAM Best Paper Runner-up Awards in 2019, and the ISCRAM Best Paper Runner-up Award in 2017. He has made significant contributions to developing AI based tools and resources (http://crisisnlp.qcri.org/) to support humanitarian organizations during disaster events and to support the UN-OCHA in streamlining their Education Insecurity effort. He is also well known in the Bangla Language computing community for his numerous contributions to advance Bangla language computing research (https://banglanlp.org/).

Tanvirul Alam is a Computing and Information Science Ph.D. student at the Rochester Institute of Technology. He received his B.Sc. degree in Electrical and Electronic Engineering from the Bangladesh University of Engineering and Technology in 2016. His current research includes information extraction from unstructured cyber threat reports, aggregating information in knowledge graphs for predictive analysis, and explainability in the context of deep neural networks. He has made several notable contributions to Bangla natural language processing, including punctuation restoration and text classification.

Ferda Ofli is a Senior Scientist at the Qatar Computing Research Institute since 2014. He received the B.Sc. degrees both in electrical and electronics engineering and computer engineering, and the Ph.D. degree in electrical engineering from Koç University, Istanbul, Turkey, in 2005 and 2010, respectively. From 2010 to 2014, he was a Postdoctoral Researcher at the University of California, Berkeley, CA, USA. His research interests cover computer vision and machine learning with applications in the humanitarian domain. He is an IEEE and ACM senior member with over 75 publications in refereed conferences and journals including CVPR, ECCV, and PAMI. He won the CVPR Outstanding Reviewer Awards in 2020 and 2021, the ISCRAM Best CoRe Paper Award in 2020, the ISCRAM Best Insight Paper and Best Paper Runner-up Awards in 2019, the ISCRAM Best Paper Runner-up Award in 2017, the Elsevier JVCI Best Paper Award in 2015, and the IEEE SIU Best Student Paper Award in 2011. He also received the Graduate Studies Excellence Award in 2010 for his outstanding academic achievement at Koç University.

Muhammad Imran is a Senior Scientist and Lead of the Crisis Computing team at Qatar Computing Research Institute. His interdisciplinary research focuses on natural language processing, social computing, computer vision, and applied machine learning. He analyzes social media communications during time-critical situations using big data analysis techniques such as data mining, machine learning, and deep neural networks. He develops novel computational models, techniques, and technologies useful for stakeholders to gain situational awareness and actionable information during sudden-onset disasters. Dr. Imran received his PhD in computer science from the University of Trento, Italy, in 2013. He then joined QCRI as a post-doctoral researcher. Dr. Imran has published over 100 research papers in top-tier international conferences and journals, including ACL, SIGIR, IJHCI, ICWSM, and WWW. Four of his papers received the ”Best Paper Award” and two ”Best Paper Runner-up Award.”