**A Comparative Study on Disaster Detection from Social Media Images using Deep Learning**

Arif, Abdullah Omar, Sabah Ashraf, AKM Mahhubur Rahman, Amin Ahsan Ali, M Ashraful Amin

Computer Vision and Cybernetics group, Dept. Of Computer Science and Engineering, Independent University, Bangladesh (IUB), Dhaka, Bangladesh

{arif1881996, mdabdullahomar97, sabah.ashraf97}@gmail.com

{akmmrahman, aminali, aminmdashraful}@iub.edu.bd

**Abstract:** Availability of images of events almost in real-time on social media has prospect in many application development. A humanitarian technology for disaster type and level assessment can be developed using the images and video available on the social media. In this paper we investigate potential use of various available deep learning techniques to develop such application. For our research, based on the use of publicly available image data, we have started collecting disaster images from various sources from South Asia. We created South Asia Disaster (SAD) image dataset containing 493 images from various online news portal. Using the keras as our framework to run our models: Visual Geometry Group (VGG-16, VGG-19), Inception-V3, Inception-Resnet-V2. However, to boost up the training speed we dropped the fully connected layer and added a small fully connected model. To identify the five different disasters: fire disaster, flood disaster, human disaster, infrastructure disaster, nature disaster; our proposed method with VGG-16 models recognition accuracy was 83.37% on SAD dataset.

**Keywords:** Disaster image, humanitarian technology, standard disaster dataset, Convolutional Neural Network, VGG-16, VGG-19, Inception-V3, Inception-ResNet-V2, Keras model.

1. **Introduction**

Bangladesh is one of the most vulnerable countries to climate change, as well as disaster prone. Bangladesh’s flat topography, low-lying and climatic features, combined with its population density and socio-economic environment, make it highly susceptible to many natural hazards, including floods, droughts, cyclones and earthquakes. Also there are disasters happening like infrastructure or non-natural disasters. Day after day people are suffering from different kinds of disasters (e.g. Fire incidents in Chawk Bazar, Banani Fire incident). These incidents which happened in March, 2019 were featured by some newspapers such as the famous online news portal Dhaka Tribune [12], The Daily Star [13] etc. It is crucial in times of crisis that how emergency response workers reach all those affected in a timely manner. It would be great to have a system that would raise an alert and assess the degree of damage of any disaster and inform appropriate authorities based on the automated analysis of the image data that are almost all the time available in realtime on various social media. There have been some effort along this direction.

Rizk et al. [1] proposed a multi-modal approach to automate crisis data analysis using machine learning. The proposed multi-modal two stage framework relies on computationally inexpensive visual and semantic features to analyze Twitter data. Level I (one) classification consists of training classifiers separately on semantic descriptors and combinations of visual features. These classifiers’ decisions are aggregated to form a new feature vector to train the second set of classifiers in Level II (two) classification. In this second classification phase, two approaches are compared: score learning and majority vote.

In [2] presents the algorithms that CERTH team deployed in order to tackle disaster recognition tasks. Deep Convolutional Neural Network (DCNN), DBpedia Spotlight and combMAX was implemented to tackle DIRSM. The model GoogleNet was used to trained on 5055 ImageNet concepts.

Giannakeris et al. [3] presented a novel warning system framework for detecting people and vehicles in danger. The proposed framework provides a near real-time localization solution for detecting and scoring severity safety levels of people and vehicles in flood and fire images. They chose to fine-tune the pre-trained parameters of the VGG-16 on Places365 dataset so as to leverage useful distinctions between various visual clues that relate to generic scenery images. Their Initial 3F-emergency dataset is composed of 6K images from Flickr.

Alam et al. [4] proposed a work where their Image Filtering module employs deep neural networks and perceptual hashing techniques to determine whether a newly arrived image is relevant for a given disaster response context. To train the relevancy filter, 3,518 images were randomly selected from the severe and mild categories. They adopted a transfer learning approach where they have used the VGG-16 network pre-trained on ImageNet data.

Mouzannar et al. [5] proposed a multimodal deep learning framework to identify damage related information from social media posts. This framework combines multiple pretrained unimodal convolutional neural networks that extract features from raw texts and images separately. The inception convolutional neural network (CNN) model was adopted which was pre-trained on ImageNet to process images and for words they used a pre-trained word embedding model to process the texts. The framework was evaluated on a homegrown labeled dataset of multimodal social media posts.

In the paper [6], images posted on social media platforms during natural disasters to determine the level of damage caused by the disasters are analyzed. In this study, Imran et al. used the VGG-16 network trained on the ImageNet dataset. They collected data from the Web (such as Typhoon Ruby, Hurricane Matthew, Nepal Earthquake etc.) and made their own dataset.They ran tests on their training sets and the highest accuracy, they achieved was when they combined their Google, Ruby and Matthew datasets.

In this research paper we propose a deep learning based method to automate the effective extraction of information from social media posts to direct relief resources efficiently. Since posts in social media contain text, photos and videos, we are performing a deep learning framework for multimodal identification of damage-related information. Also our objective is to collect and work with South Asian disaster images which include disaster images of Bangladesh, and similar countries.

1. **Methodology**

Our proposed system uses keras as framework to implement our models and modified deep learning algorithms to classify five different types of disaster images. And for all deep learning based methods data is the most important component.

* 1. **Dataset Description**

The main issue using a deep learning system for classificaton is a well defined and diversified dataset. For this experiment we collected the images from Kaggle [11]. There are six different damage categories there are, Fire damage, Flood damage, damage infrastructure, damage nature, Human damage and Non-damage. There are total 5885 images and total number of images for each category is given in Table 1.

**Table 1.** Distribution of number of disaster images for each class from Kaggle Dataset [11]

|  |  |
| --- | --- |
| Categories | Number |
| Fire damage | 349 |
| Flood damage | 385 |
| Damage Infrastructure | 515 |
| Damage nature | 1418 |
| Human damage | 240 |
| Non damage | **2972** |
| **Total** | **5885** |

The fire disaster image that contains fire element, smoke and burning down objects. Flood disaster images contains water and seeing objects under water. Infrastructure disaster contains broken buildings and rusted objects. Nature disaster contain broken trees, buildings and roads caused by earthquake or cyclone. Human disaster image contains bleeding, burned face, torturing people and human damaging the environment. Non – disaster images contain products, cosmetics, books, human models and people eating food. Some sample images from the Kaggle data set is given in Fig. 1. From Table 1 we can observe that Non damage has the highest number of images 2972 than other classes, the lowest number of images are human damage 240. Damage nature has the second highest number of images 1418. Other classes have adjacent number of images.





**Fig. 1.** Sample images from Kaggle Dataset (left-right): fire disaster, flood disaster, human disaster, infrastructure disaster, nature disaster, and non-disaster.

In the sample images note that the images are from mostly non south asian. We intend to implement our system for Bangladesh and to check the performance on the native data we collected a small data set for this project. We are calling it, **SAD (South Asian Disaster)** image dataset. We have collected 493 data and class wise distribution can be found in Table 2.

**Table 2.** Distribution of number of disaster images for each class from SAD Dataset

|  |  |
| --- | --- |
| Categories | Number |
| Fire damage | 82 |
| Flood damage | 82 |
| Damage Infrastructure | 65 |
| Damage nature | 73 |
| Human damage | 81 |
| Non damage | **110** |
| **Total** | **493** |

In Fig. 2, we gave samples of SAD image dataset. Images where collected from online news portal, social news network, independent news organization and online benefactor for people. We got 32 images from each online news portal such as the Times of India, The Hindu and Indian express. 32 images are collected from UN news (news portal), 32 images from World Bank (online benefactor for people), 32 images from Rappler (social news network), 100 images from BBC (online news portal), 100 images from Fox news (online news portal) and rest from Al Jazeera (independent news organization). Visual Comparison among images from Fig. 1 and Fig. 2 it is clear that there are differences in perception of similar classes.

  

  

**Fig. 2.** Sample images from SAD image dataset (left-right): fire disaster, flood disaster, human disaster, infrastructure disaster, nature disaster, and non-disaster

* 1. **Convolutional Neural Networks (CNNs)**

Collected data were used to train Convolutional Neural Networks. CNNs have shown state-of-the art efficiency in a multitude of computer vision functions including image classification and retrieval, object detection, and segmentation of images [10]. Their achievement in such a broad range of implementation is ascribed to their hierarchical where they automatically learn discriminatory characteristics from raw information. A typical CNN comprises of various processing layers, including convolution, pulling, and fully connected layers. These layers are structured so that one layer’s output becomes the next layer’s input. A set of kernels are introduced to the input information to produce characteristic maps at each convolution layer [10]. In [3] and [6] VGG-16 CNN model was implemented. The [5] Inception-V3, Inception-V4, VGG-16 and Inception-Resnet-V2 were used for disaster image classification. For our problem, we used VGG-16, VGG-19, Inception-V3 and Inception-ResNet-V2.

**Visual Geometry Group (VGG) [8]**: The VGG nets were originally designed to detect and recognize objects in images. VGG Nets proved to be efficient in identification of objects in still pictures. VGG is very profound model that significantly enhanced a broad variety of visual recognition functions, including detecting objects, semantic segmentation, picture captioning, and recognition of action on video. VGG-16 has three fully connected layers follow a stack of convolutional layers, the final layer is the softmax layer. It has a total of 16 layers; all hidden layers are equipped with rectification. VGG-19 model is very similar to the VGG-16 model. The main difference between VGG-16 and VGG-19 is that the VGG-19 has 19 layers instead of 16.

**Inception [9]:** Though, VGG Net has architectural simplicity and convincing features; this comes at a high price of relatively large computation time to evaluate the network. Inception architectures of GoogLeNet was intended to perform well even under rigorous memory and computer budget limitations. Inception's computational cost is also much smaller than VGG Net. This made it possible to use Inception networks in large-data situations where large amounts of data required to be processed at reasonable price or where memory or computing capability are inherently restricted. Inception-v1 has 9 inception modules stacked linearly, and 27 layers deep including pooling layers. In Inception-V2 the 5x5 convolution was factorized to two 3x3-convolution operations to improve computational speed and reduce computational expense. The Inception-V3 (INv3) is 42 layers deep, however, the computation is only about 2.5 times more than Inception-v1 and much more efficient than VGG Net.

**Inception-ResNet-V2 [7]:** Inception-ResNet-V2 (INRv2) is a 164 profound layer deep neural network that can classify pictures into 1000 categories of objects. Inception-ResNet-V2 has introduced residual connections that add the output of the convolution operation of the inception module to the input. For residual addition to work, the input and output after convolution must have the same dimensions. The pooling operation inside the main inception modules were replaced in favor of the residual connections. To increase stability, the residual activations were scaled by a value within 0.1 to 0.3 range. It was found that Inception-ResNet models were able to achieve higher accuracies at a lower number of epochs.

The models that we are using for our experiment are: VGG-16, VGG-19, Inception-V3 and Inception-ResNet-V2. In Table 3 a comparative view of number of parameters of all four above mentioned CNN models is given. However, we also implemented a small model, and added that at the end of each of the above mentioned models, this is done by dropping conventional fully connected layer at the end. The reason for adding this small model is to speed up the training process.

In Fig. 3 we gave the schematic view of the models used. We have used relu activation on the 512 dense layers and also on the 256 dense layers so that all the layers are connected to each other. And then we used dropout, it helps reduce overfitting, by preventing a layer from seeing twice the exact same pattern. At the end we used 6 dense layers because we have 6 classes. As optimizer we used Adam and as loss function sparse categorical cross entropy is used.

**Table 3.** Parameters of the models

|  |  |
| --- | --- |
| Models | Parameters |
| VGG 16 | 138 Millions |
| VGG 19 | 144 Millions |
| Inception-V3 | 23 Millions |
| Inception-ResNet-V2 | 54 Millions |

512 Layers 256 Layers Dropout 6 Layers

VGG-19/ VGG-16/

Inception-V3/

InceptionResnet-V2

**without last layer**









Relu Relu Sigmoid

Proposed modification module

**Fig.3.** Schematic view of the proposed deep neural model

1. **Experimental Setup**

Images are usually of different size as they are collected from non-restricted sources. To feed them into the deep neural networks they were resized to 150 by 150 pixels. For training the network 50 epochs have been used, epoch basically means training the whole dataset in one cycle, however it takes a lot of space in the memory. Therefore, the dataset is divided into batch to train. The batch size 11 and 12 are used for Kaggle training and testing dataset respectively. For SAD dataset the batch size is 29 for the whole dataset. To run this experiment, we are using intel core i5 processor and 8 GB DDR4 ram. The experiments that we are performing are:

1. **Experiment 01:** Cross validation performance measure while training the CNNs with Kaggle dataset testing them on Kaggle dataset. Table 4 shows the results for this experiment.
2. **Experiment 02:** Performance measure while training with 100% Kaggle dataset and test with 100 % SAD dataset. Table 5 shows the results for this experiment
3. **Experiment 03:** Cross validation performance measure while training with 100% Kaggle dataset plus 60% of the SAD dataset and test with the remaining 40% of SAD dataset. Table 6 shows the results for this experiment.

In Table 4, note that all four classifiers perform in the same range on average. However, different classifiers show slightly higher classification accuracy class of disaster.

**Table 4.** Cross-validation average performance of only Kaggle dataset

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Precision (%)** | | | | **Recall (%)** | | | | **Accuracy(%)** | | | | |
|  | **VGG19** | **VGG16** | **INv3** | **INRv2** | **VGG19** | **VGG16** | **INv3** | **INRv2** | **VGG19** | **VGG16** | **INv3** | **INRv2** |
| **Fire damage** | 63.31 | 58.74 | **64.44** | 58.22 | 68.00 | 67.33 | 67.96 | **75.66** | 95.51 | 95.62 | 95.72 | **96.08** |
| **Flood damage** | **59.40** | 51.63 | 41.93 | 51.12 | 54.82 | 60.64 | **63.32** | 63.12 | 93.92 | **94.37** | 93.91 | 94.25 |
| **Human damage** | 57.63 | **59.11** | 49.92 | 56.00 | **71.21** | 61.32 | 69.00 | 69.23 | **97.07** | 96.54 | 96.82 | 96.79 |
| **Damage Infrastructure** | **70.92** | 70.12 | 63.33 | 69.84 | 65.13 | 63.55 | **67.66** | 67.16 | 83.87 | 83.04 | 83.54 | **84.33** |
| **Damage Nature** | 37.75 | 47.00 | 48.10 | **49.33** | **52.95** | 51.00 | 47.91 | 50.94 | **91.28** | 91.13 | 90.40 | 91.06 |
| **Non-damage** | 83.52 | 82.71 | 85.55 | **85.82** | **83.25** | 83.13 | 79.84 | 81.83 | **83.65** | 82.91 | 81.79 | 82.60 |
| **Average** | 62.08 | 61.55 | **67.20** | 61.72 | 65.89 | 64.50 | 65.94 | **67.99** | **90.88** | 90.60 | 90.36 | 90.85 |

In Table 5 first issue to note is that, as we test the classifiers trained with Kaggle dataset and test with our SAD dataset the accuracy reduces by about 7%. This means that the Kaggle dataset does not contain well distributed diverse data from all demography and SAD data is different from Kaggle dataset (Fig. 4). Something very noticeable in this experiment is that, the precision for human damage is significantly low for all classifiers (Fig. 1 and Fig. 2). This would mean that, identifying human damage by a deep neural net trained with Kaggle dataset only is very difficult. VGG16 is an older version of deep neural network; however, overall accuracy of this classifier is slightly higher than other models for experiment 2.

**Table 5.** Performance of training with 100% Kaggle dataset and testing with 100% SAD dataset

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Precision (%)** | | | | **Recall (%)** | | | | **Accuracy(%)** | | | |
|  | **VGG19** | **VGG16** | **INv3** | **INRv2** | **VGG19** | **VGG16** | **INv3** | **INRv2** | **VGG19** | **VGG16** | **INv3** | **INRv2** |
| **Fire damage** | 46.33 | 59.73 | **60.92** | 37.83 | 97.44 | 83.11 | 81.92 | 100 | 90.87 | **91.28** | **91.28** | 89.66 |
| **Flood damage** | 45.12 | **59.73** | 46.33 | 43.93 | 74.02 | 62.83 | 80.52 | 75.00 | 88.24 | 87.42 | **89.25** | 88.24 |
| **Human damage** | 6.15 | 9.23 | 9.23 | **10.73** | 44.44 | 33.33 | 35.33 | 43.71 | **86.61** | 85.60 | 85.50 | 86.41 |
| **Damage Infrastructure** | **82.22** | 75.33 | 79.52 | 68.52 | **32.11** | 36.63 | 26.92 | 30.11 | 71.60 | 77.08 | 64.91 | **71.81** |
| **Damage Nature** | 30.83 | 32.12 | 17.31 | **51.82** | 56.81 | 54.23 | 51.82 | 51.22 | **84.79** | 84.38 | 83.77 | 83.98 |
| **Non-damage** | **80.00** | 71.82 | 61.83 | 70.00 | 53.71 | **56.44** | 54.44 | 51.34 | 80.12 | **81.34** | 79.92 | 78.51 |
| **Average** | 48.44 | **51.32** | 45.85 | 47.13 | **59.75** | 54.42 | 55.15 | 58.56 | 83.70 | **84.51** | 82.43 | 83.10 |

In experiment 3, we added 60% of the SAD data with the Kaggle dataset and it shows the improvement in performance. In Table 6, note that the precision of human damage improves. Moreover, accuracy is also increased. It suggests that if the networks can be trained with significantly large diverse data, performance will increase with the same network setup.

**Table 6.** Cross-validation performance of training with 100% Kaggle dataset plus 60% SAD dataset, and testing with 40% SAD dataset

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Precision (%)** | | | | **Recall (%)** | | | | **Accuracy(%)** | | | |
|  | **VGG19** | **VGG16** | **INv3** | **INRv2** | **VGG19** | **VGG16** | **INv3** | **INRv2** | **VGG19** | **VGG16** | **INv3** | **INRv2** |
| **Fire damage** | **72.22** | 66.66 | 48.55 | 69.66 | 96.00 | 91.66 | 88.81 | **100** | **94.95** | 93.43 | 90.40 | 94.68 |
| **Flood damage** | 34.44 | 53.11 | 40.66 | 40.61 | **84.63** | 77.33 | 81.33 | 65.00 | 88.38 | **89.90** | 88.89 | 85.11 |
| **Human damage** | 17.43 | **26.12** | 8.69 | 21.72 | 36.44 | 35.32 | **66.66** | 38.55 | 86.87 | 85.86 | **88.89** | 84.04 |
| **Damage Infrastructure** | **87.81** | 78.82 | 84.81 | 78.73 | 39.22 | **44.12** | 40.51 | 40.00 | 75.25 | **79.80** | 79.77 | 70.74 |
| **Damage Nature** | 23.33 | 26.63% | **33.33** | 23.33 | **87.54** | 50.00 | 45.55 | 50.00 | **87.88** | 84.85 | 83.84 | 86.17 |
| **Non-damage** | 80.81 | 78.71 | **87.22** | 76.52 | 56.73 | **61.66** | 58.62 | 57.12 | 80.81 | **83.33** | 82.32 | 76.06 |
| **Average** | 52.67 | **55.01** | 50.54 | 51.76 | **66.76** | 60.02 | 63.58 | 58.45 | 85.69 | **86.20** | 85.68 | 82.80 |

Comparing table 4 and 5, the accuracy dropped around 7% when preforming the second experiment. The reason for this drop of accuracy maybe for some specific class there is significant between Kaggle and SAD dataset. In Fig. 4 we observe that SAD flooding disaster and nature disaster is different than other countries flooding disaster and nature disaster.

 

1. (b)

 

(c) (d)

**Fig. 4**. (Left – Right; Top-bottom) (a)SAD Flood damage; (b) Other Countries Flood damage; (c) SAD Damage Nature; (d) Other Countries Damage Nature

In Fig. 5 we provide some images associated with fire class, here we observed that, there are two types of images: During fire disaster and after fire disaster. In Fig. 5 not that during fire disaster and after fire disaster images are visually very distinctive. The difference is, in one has fire element and smoke in it and the other one is all black and there are ashes. Visually we ourselves are confused between middle image (after fire disaster) and right image (damaged infrastructure). Thus, the deep networks are also sometimes confused with after fire image and infrastructure disaster image.

  

**Fig. 5.** (Left - Right) (a) During Fire damage; (b) After Fire damage; (c) Damage Infrastructure.

In Fig. 6 we provide some more images that are misclassified by our system.

(a) After fire damage classified as Damaged Infrastructure; (b) Is an actual infrastructural damage training image; (c) Another after fire damage classified as flood; in image (d) the black and white floor and books floating resembles the state of the floor of (c) and in (e) the training flood image does not really look like a flood image but there are pipes appearing in it which also appears in test image (c). We believe that, some of the fire test images were classified as flood and infrastructure by Inception-V3 and Inception-ResNet-V2 models, because, the test image does not actually contain any fire object in them.

 

****  

**Fig. 6.** Wrong Predictionby proposed system; (Left-Right) (a) After fire damage classified as Damaged Infrastructure; (b) Actual training Damaged Infrastructure image; (c) Fire damage classified as flood damage;(d) and (e) Actual Flood damage training image.

1. **Conclusion**

Real time support is the key issue in minimizing damage by any disaster; most of the time traditional method of assessing need for support in case of disaster takes too long and in the meantime the damage has been done. In this paper we propose a method to determine the type of disaster automatically from social media images. To test the performance of the proposed idea publicly available dataset collected from Kaggle is used. Moreover, we extended the dataset by gathering 493 disaster images of six classes from this region and we are calling it “South Asia disaster” (SAD). Our experiments revealed that the Kaggle dataset is not diverse and does not perform well recognizing disaster in SAD images. We implemented four different types of deep neural network models: VGG 16, VGG 19, Inception-V3 and Inception-ResNet-V2, for this research. However, to boost up the training speed we removed the fully connected layer and added a small fully connected model. To identify the five different disasters: fire disaster, flood disaster, human disaster, infrastructure disaster and nature disaster; our proposed method with VGG 16 models recognition accuracy is 83.37% on SAD dataset.

**Reference**

1. Y. Rizk, M. Awad, and C. Castillo, “A Computationally Efficient Multi-modal Classification Approach of Disaster- related Twitter Images.”
2. K. Avgerinakis, A. Moumtzidou, S. Andreadis, E. Michail, I. Gialampoukidis, S. Vrochidis, and I. Kompatsiaris, “Visual and textual analysis of social media and satellite images for flood detection @ multimedia satellite task MediaEval 2017,” pp. 3–5, 2017.
3. P. Giannakeris, K. Avgerinakis, A. Karakostas, S. Vrochidis, and I. Kompatsiaris, “People and vehicles in danger - A fire and flood detection system in social media,” 2018.
4. F. Alam, M. Imran, and F. Ofli, “Image4Act : Online Social Media Image Processing for Disaster Response,” pp. 601–604, 2017.
5. H. Mouzannar and M. Awad, “Damage Identification in Social Media Posts using Multimodal Deep Learning,” no. May, 2018.
6. D. T. Nguyen, F. Ofli, M. Imran, and P. Mitra, “Damage Assessment from Social Media Imagery Data During Disasters,” 2017.
7. C. Szegedy, G. Inc., C. 1600 Amphitheatre Pkwy, Mountain View, Szegedy@google.com, S. Ioffe, Sioffe@google.com, V. Vanhoucke, Vanhoucke@google.com, A. Alemi, and Alemi@google.com, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.”
8. X. Zhang, J. Zou, K. He, and J. Sun, “Accelerating Very Deep Convolutional Networks for Classification and Detection,” pp. 1–14.
9. C. Szegedy, V. Vanhoucke, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” 2014.
10. K. Muhammad, J. Ahmad, and S. W. Baik, “Early fire detection using convolutional neural networks during surveillance for effective disaster management,” *Neurocomputing*, vol. 0, pp. 1–13, 2018.
11. https://www.kaggle.com/
12. https://www.dhakatribune.com/
13. https://www.thedailystar.net/