Disaster Damage Identification in Social Media Posts Final Report

Data Science Capstone

Yuan Meng

November 27, 2021

Table of Contents

1. Introduction

a. Introduction/Background

Social media content has become a vital information source for rescuers when significant disasters happen. Reports of injured people, infrastructure damage, and missing people on social media provide timely and specific information to help rescue organizations rapidly locate the place or people that need help.

b. Problem Statement

The rescue staffing cannot deal with all the dangers immediately when major disasters occur, so it is necessary to grade the situation according to the urgency. This project aims to apply a convolutional neural network model to identify the pictures with disaster types and the degree of urgency to help rescuers arrange rescues reasonably.

c. Problem Elaboration

First, how does the model work? The more serious the disaster in the picture, the higher the emergency degree of rescue. The collected data has marked disaster categories and severity, and the model uses these as labels for training. The CNN model first identifies the type of disaster and the urgency (degree of disaster). Second, how can the model work for social media posts? The model can be used for social media posted images with disasters tags. For example, when a city faces a flood, hundreds of rescue messages with flood photos may appear on social media within one minute. However, rescuers cannot go to these locations simultaneously. With this model for urgency classification, rescuers can prioritize those areas that have been severely affected and reduce casualties.

d. Motivation

My personal experience inspired the idea of this project. This summer, the city I lived in was affected by severe flooding due to a long time of heavy rainfall. Almost the whole town suffered water and power outages for a week, and the helplines were busy for 24 hours every day. There were thousands of help posts on social media such as Weibo, and many of them got resolved. However, manually dealing with those messy contents could be inefficient and unorganized, so I want to create a project that automatically collects the relative information and organizes it.

e. Project Scope

The ideal project consists of four parts: first, it automatically gathers target tweets based on related tags; second, it identifies the degree of urgency using image classification; third, it verifies if the post contains enough information such as location and contact details; and at last, it displays the organized results. While due to the time limitation of the work, I only focused on the image classification part.

2. Literature Review

a. Relevant Research

The first study (Alam et al., 2018) indicates extensive research uses social media textual content but little focuses on images shared on social media due to the lack of labeled imagery data. To solve the issue, it introduces CrisisMMD datasets with three types of annotations: informativeness, humanitarian categories, and damage severity categories. The research also proves the dataset can fulfill several humanitarian use cases and tasks.

The second research (Mozannar et al., 2018) informs that an automated system that detects and flags social media posts for disaster and crisis data is essential to efficiently direct relief resources because monitoring social media content becomes common among emergency responders. And it focuses on the second part of the automated systems, identifying infrastructure and environmental damage elements in posts.

The third study (Ofli et al., 2020) uses multimodal deep learning techniques to analyze the social media data for disaster response. It uses both text and image modalities of Twitter data to learn informative and humanitarian tasks, such as if a tweet contains informative content or shows affected individuals or infrastructure damage.

3. Methodology

a. Dataset Description

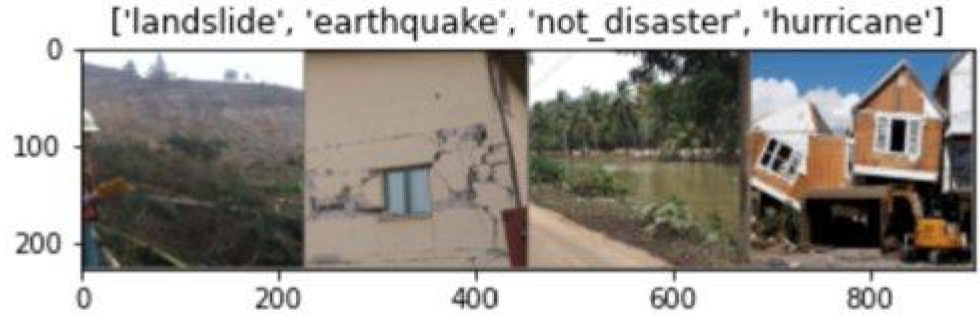
The crisis image benchmark dataset (Alam et al., 2020) contains thousands of labeled images collected from Twitter with different disasters and three damage severity levels. Disaster types include earthquake, fire, flood, hurricane, landslide, not disaster, and other disasters, with severe damage: little or none, mild, or severe. Although it also includes the labels of informativeness and humanitarian classes, the model didn’t use these categories because it only aims to classify the severity of the damage. In addition, each type of data contains train, validation, and test sets.

b. Data Collection

The dataset has its official website, so I directly downloaded it using the “wget” method and unzipped it into the environment using “os” method in Python.

c. Data Preprocessing

Firstly, I built a custom dataset class and data loader using the PyTorch dataset to read the image files, encode labels, define the batch size and shuffle them. Second, I deleted the “other disaster” class because it combines various images and could mislead the model. Third, I resized and normalized the image data and converted them to tensors using the transform functions. I also randomly cropped and randomly flipped images for data augmentation for the training dataset.



Here are some sample data after processing.

d. Data Modeling & Visualizations

Pre-trained networks have sufficient training datasets and reasonable architectures, so I decided to use the transfer learning method for this project. As introduced in the literature review, resnet 50, vgg 16, and inception v3 are suitable for disaster image classification. I loaded these models and modified the final linear layer with the number of features and labels of the datasets.

I used the CrossEntropyLoss function in the PyTorch library for the loss function, and the sum of the output was divided by the number of elements in the output. The loss function is:

$$\text{loss}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right).$$

Also, I used SGD optimizer with 0.9 momenta, the network parameters, and the learning rate change while training.

To train the network, I used a learning rate scheduler: start with an initial learning rate of 0.01, and it decays the learning rate by a factor of 0.1 every for every 7 epoched, with a total 15 epochs for training. Every batch of training and deviation data is loaded separately for every epoch. The model zeros the parameter gradients and computes the running loss and accuracy. Only the training step tracks the forward loss history, calculates the backward loss, and optimizes it. Then the model adds the total loss and accuracy for each epoch, and if the accuracy is higher than the record, the best model weights will be updated.

After training and fine-tuning the three models, I found inception v3 works best for my datasets. So, here is the architecture of inception v3.

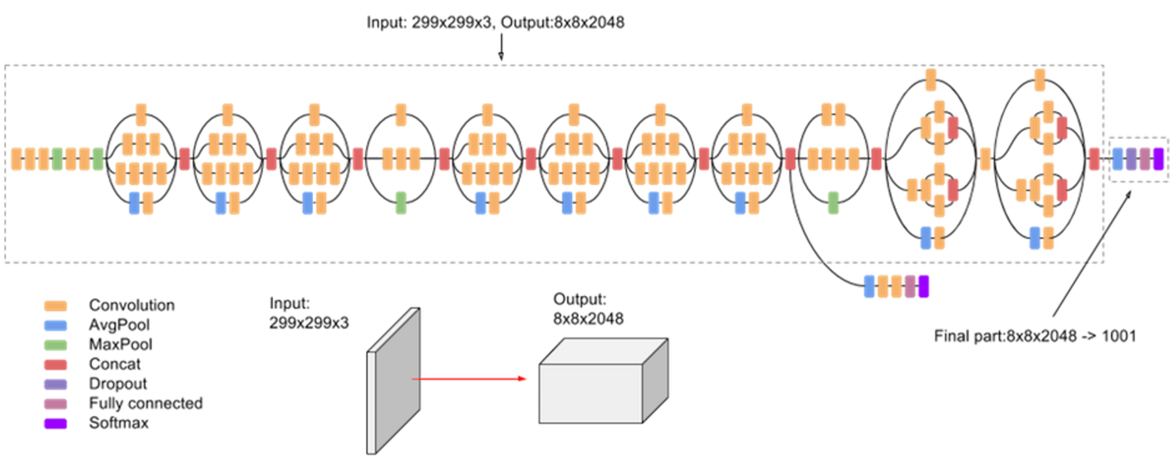


Figure from paperswithcode

Inception-v3 is a convolutional neural network architecture proposed in the paper Rethinking the Inception Architecture for Computer Vision (Szegedy et al., 2015). Based on the exploration of ways to scale up networks to utilize the added computation as efficiently as possible by suitably factorized convolutions and aggressive regularization.

The factorized convolutions help to reduce the computational efficiency as it reduces the number of parameters involved in a network. However, it also keeps a check on the network efficiency. It also has smaller convolutions: replacing bigger convolutions with smaller convolutions leads to faster training. Then, an auxiliary classifier propagates label information lower down the network (along with the use of batch normalization for layers in the sidehead). In addition, it has grid size reduction to combat the bottlenecks of computational cost.

4. Results & Analysis

```
Epoch 19/19
----------
train Loss: 0.4163 Acc: 0.8631
dev Loss: 0.4503 Acc: 0.8531

Training complete in 135m 38s
Best val Acc: 0.853052
```

	precision	recall	f1-score	support
earthquake	0.88	0.81	0.84	404
fire	0.89	0.90	0.89	280
flood	0.86	0.89	0.88	599
hurricane	0.76	0.69	0.73	352
landslide	0.81	0.85	0.83	268
not_disaster	0.90	0.93	0.91	990
accuracy			0.86	2893
macro avg	0.85	0.84	0.85	2893
weighted avg	0.86	0.86	0.86	2893

As the graphs show above, I got 0.85 validation accuracy and 0.86 test accuracy for the disaster type data set. The not disaster category has a 0.91 f1 score, so the model could primarily identify whether one image represents disasters.

```
train Loss: 0.3499 Acc: 0.8644
dev Loss: 0.6863 Acc: 0.7489

Training complete in 232m 9s
Best val Acc: 0.754425
```

	precision	recall	f1-score	support
little_or_none	0.84	0.87	0.86	2135
mild	0.44	0.37	0.40	629
severe	0.74	0.75	0.74	1101
accuracy			0.76	3865
macro avg	0.67	0.66	0.67	3865
weighted avg	0.75	0.76	0.75	3865

And the damage severity dataset didn't get results as good as the last one. It’s hard for the model to identify the mild damages correctly. It may be because I combined different types of disasters to define the damage severity.

5. Conclusion

a. Conclusion

In conclusion, my project aims to identify the gathered social media pictures with disaster types and the degree of urgency to help rescuers arrange rescues reasonably. I applied the inception v3 model for both disaster type dataset and damage severity dataset to classify them and got about 0.85 accuracy.

b. Project Limitation

The limitation of the project is I could not train for the selected disaster type to identify their damage severity because the data has different images for different tasks, so I could only use the various types of disasters to define the damage severity.

c. Future Research

For future research, I’ll collect only one type of disaster image from Twitter and label them by myself to re-train the model to see any improvements. Also, I will work on some other models to improve the results, such as the MLP-Mixer architecture.

6. References

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

Firoj Alam, Ferda Ofli, Muhammad Imran, Tanvirul Alam, Umair Qazi, Deep Learning Benchmarks and Datasets for Social Media Image Classification for Disaster Response, In 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2020.

Firoj Alam, Ferda Ofli, and Muhammad Imran, CrisisMMD: Multimodal Twitter Datasets from Natural Disasters. In Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM), 2018, Stanford, California, USA.

Ferda Ofli, Firoj Alam, and Muhammad Imran, Analysis of Social Media Data using Multimodal Deep Learning for Disaster Response, In Proceedings of the 17th International Conference on Information Systems for Crisis Response and Management (ISCRAM), 2020, USA.

Hussein Mozannar, Yara Rizk, and Mariette Awad, Damage Identification in Social Media Posts using Multimodal Deep Learning, In Proc. of ISCRAM, May 2018, pp. 529–543.