KLE Society's

KLE Technological University

December 21, 2021

Course Project
On
Detecting Natural Disasters

Machine Learning(17ECSC306)

Submitted by

Name	Roll No	USN
Muskan Havaldar	164	01FE19BCS007
Shubham Khobare	142	01FE19BCS047
Akhilesh Naik	158	01FE19BCS066
Rohit Hegade	151	01FE19BCS058

Team Number: A13

Under The Guidance Of
Dr. Meena S M
Mr. Uday Nagraj Kulkarni

SCHOOL OF COMPUTER SCIENCE ENGINEERING
HUBLI-580031
Academic year 2021-2022

Abstract

A natural disaster is a major adverse event resulting from natural processes of the Earth; examples include firestorms, floods, hurricanes, tornadoes, volcanic eruptions, earthquakes, tsunamis and other geologic processes. A natural disaster can cause loss of life or damage property, and typically leaves some economic damage in its wake, the severity of which depends on the affected population's resilience and on the infrastructure available. As a part of our course project, we explore the possibility of machine learning approaches for scalable and early natural disaster detection. We used the dataset named 'CycloneWildfireFloodEarthquakesDataset', a dataset available for natural disaster detection.

The CycloneWildfireFloodEarthquakesDataset consists of about 6,828 images of natural disasters collected which spans over 4 classes. We tried implementing with our pre-trained model CNN model and we discussed the results. We also build an application and deployed it on android machines which can help in detecting natural disasters. Overall, the approach of training deep learning models on increasingly large and publicly available image datasets presents a clear path towards smartphone-assisted detecting natural disasters on a massive global scale.

Acknowledgement

Every project is successful largely due to the effort of a number of people who have always given their valuable advice or lent a helping hand. We sincerely appreciate the inspiration, support and guidance of all those people who have been instrumental in making this project a success.

We take this opportunity to thank Dr.Ashok Shettar, Vice-chancellor,KLE Technological University and to Dr. Prakash Tewari, Principal, B V Bhoomaraddi College of Engineering and Technology, Hubli.

We take this opportunity to thank Dr.Meena S M, Head, School of Computer Science and Engineering for having provided us with an academic environment which nurtured our practical skills contributing to the success of our project.

We also take this opportunity to thank Prof. Uday Kulkarni for the guidance and constant supervision as well as for providing necessary information regarding the project & also for supporting us throughout the project. We are thankful and fortunate enough to get constant encouragement, support and guidance from all teaching and non teaching staff of SoCSE which helped us in successfully completing our project.

We owe our gratitude to the School of Computer Science Engineering, KLE technological University for providing us with the resources necessary for the completion of this project. Our gratitude will not complete without thanking the Almighty, our beloved parents, our seniors and our friends who have been a constant source of blessings and aspirations.

Contents

1 Introduction

1.1 Overview of the project

Natural disasters are catastrophic events with atmospheric, geological, and hydrological origins (e.g., droughts, earthquakes, floods, hurricanes, landslides) that can cause fatalities, property damage and social environmental disruption. Natural disasters are common, devastating and impact a large number of people annually. Natural disasters are out of human control but the consequences of natural disasters overlap with the consequences of war or combat. In both contexts, there is human suffering caused by damage to life, personal property, and infrastructure. Families are displaced and victims lose shelter. This is complicated further by immense shortages of food and drinking water.

With billions of smartphones around the globe, wouldn't it be great if the smartphone could be turned into a natural disaster detecting tool, recognizing disasters from images it captures with its camera? This challenge is the first of many steps turning this vision into a reality and as a part of our course project, our aim is to develop learning model than can accurately detect natural disaster based on an image.

2 Objectives

1) To detect the natural disasters caused in the world.
2) Classification of natural disasters using texture features.
3) Build an application using the trained model to predict the class of the natural disaster image.

2.1 Literature survey

2.1.1 Image Classification for Content-Based Indexing

Grouping images into (semantically) meaningful categories using low-level visual features is a challenging and important problem in content-based image retrieval. Using binary Bayesian classifiers, we attempt to capture high-level concepts from low-level image features under the constraint that the test image does belong to one of the classes. Specifically, we consider the hierarchical classification of vacation images; at the highest level, images are classified as indoor or outdoor; outdoor images are further classified as city or landscape; finally, a subset of landscape images is classified into sunset, forest, and mountain classes. We demonstrate that a small vector quantizer (whose optimal size is selected using a modified MDL criterion) can be used to model the class-conditional densities of the features, required by the Bayesian methodology. The classifiers have been designed and evaluated on a database of 6931 vacation photographs. Our system achieved a classification accuracy of 90.595.3and 96develop a learning method to incrementally train the classifiers as additional data become available. We also show preliminary results for feature reduction using clustering techniques. Our goal is to combine multiple two-class classifiers into a single hierarchical classifier

2.1.2 Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units

Recently, convolutional neural networks (CNNs) have been used as a powerful tool to solve many problems of machine learning and computer vision. In this paper, we aim to provide insight on the property of convolutional neural networks, as well as a generic method to improve the performance of many CNN architectures. Specifically, we first examine existing CNN models and observe an intriguing property that the filters in the lower layers form pairs (i.e., filters with opposite phase). Inspired by our observation, we propose a novel, simple yet effective activation scheme called concatenated ReLU (CReLU) and theoretically analyze its reconstruction property in CNNs. We integrate CReLU into several state-of-the-art CNN architectures and demonstrate improvement in their recognition performance on CIFAR-10/100 and ImageNet datasets with fewer trainable parameters. Our results suggest that better understanding of the properties of CNNs can lead to significant performance improvement with a simple modification

2.2 Problem definition

The goal is to classify the disaster using Cyclone Wildfire Flood Earthquake Dataset, deep learning algorithms can be used to automatically detect natural disasters in images.

3 Approach

3.1 Dataset

The dataset is named as ”CycloneWildfireFloodEarthquakes-Dataset”.The dataset consists of about 6050 images of natural disasters been detected.We analyze 6828 images of natural disasters, which have a spread of 4 class labels assigned to them namely Cyclone, Wildfire, Floods and Earthquakes.

Each class label represents a natural disaster, and we make an attempt to detect the natural disaster given just the image of the disaster.

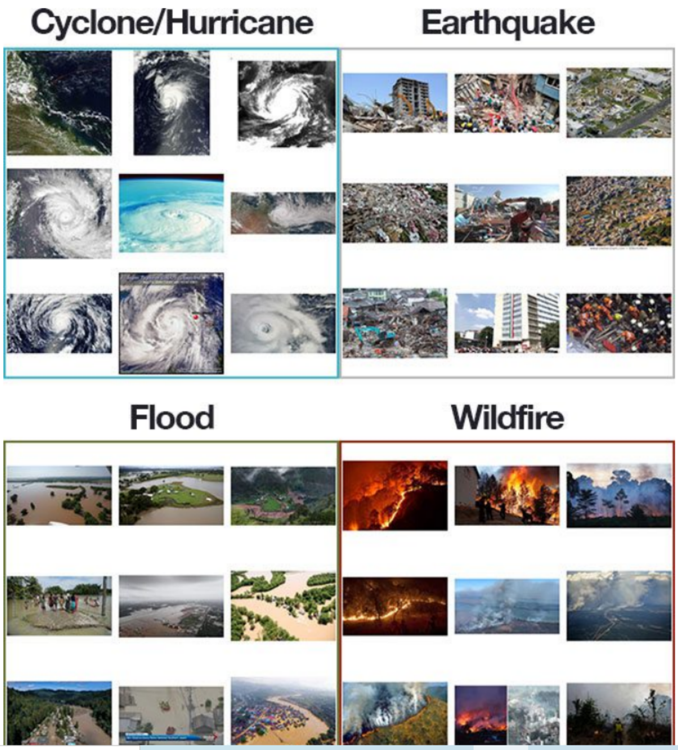


Figure 1: All The Classes of Natural disaster in Dataset

Figure 1 shows sample images of every natural disaster from the CycloneWildfireFloodEarthquakesDataset.

3.2 Methodology

Our Methodology consists of 3 phases, Pre-processing, Model Training, App Building. The images in the data-set are Pre-processed i.e. resizing the images, increasing data-set by image augmentation, so that the model will be robust and will be able to classify even in realtime.

In first phase, the images are split into train, test and validation datasets, also resize and normalize the input to the same format. In second phase, i.e. model training, the model is trained on the pre-processed data. This model is saved and used for building the app in the third phase.

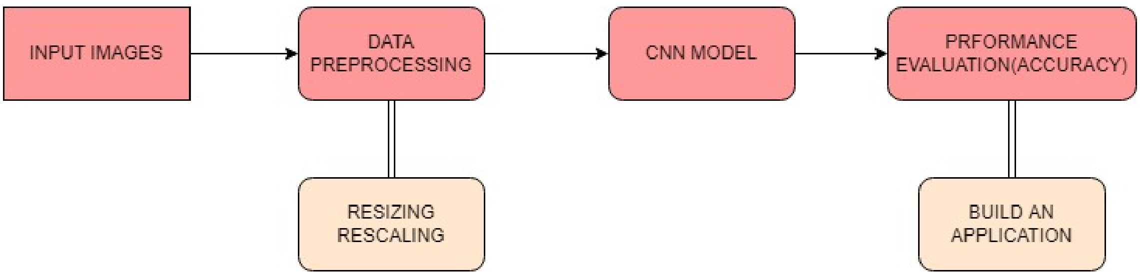


Figure 2: Methodology

4 A Deep Walk Through The Pipeline

4.1 Preprocessing

The images were copied into train, test and validation folder as per their class labels using os and shutil libraries. Each class of: train contains 400 images, test contains 100 images and validation contains 100 images. Then to normalize the data the images were converted to a default size(224,224), mode is set to 'RGB'(triple 8-bit value for red, green and blue), this is done using python imaging library(PIL). Then Image data generator is imported from Keras for implementing data augmentation on the data. The process of data augmentation provides diversity to the data and also makes the deep learning model robust. So that the model does not have to depend on only clear and correct data all the time.

```
# summarize some details about the image
print(image.format)
print(image.mode)
print(image.size)
plt.imshow(image)
```

```
JPEG
RGB
(224, 224)
```

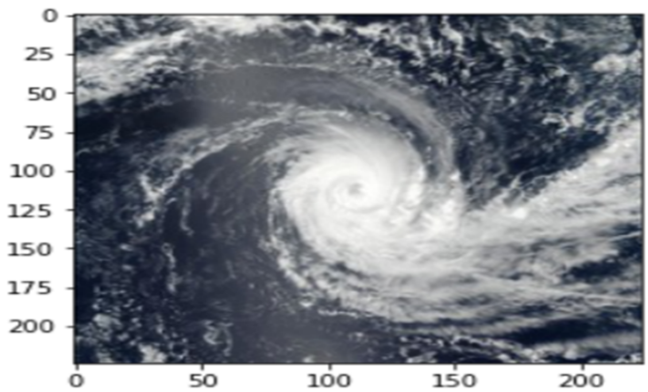
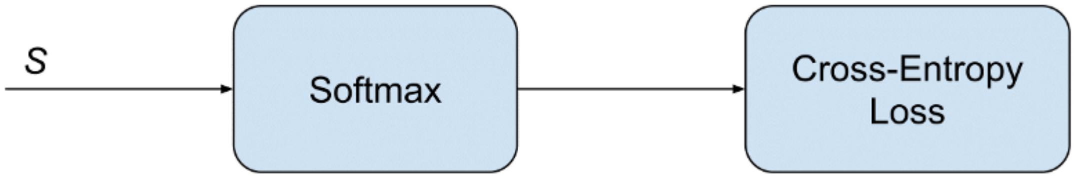


Figure 3: Details of an image from cyclone directory

4.2 Loss Function

Here we are using categorical cross-entropy as loss function to minimize. It is also called as Softmax Loss. It is a Softmax activation plus a Cross-Entropy loss. If we use this loss, we will train a CNN to output a probability over the C classes for each image. It is used for multi-class classification.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

Figure 4: Cross Entropy

4.3 Model Design

Our Model consists of 4 convolution blocks followed by relu activation functions also batch normalization blocks. The first Convolution block has 32 output filters with kernel size (3,3) and activation function applied is relu. MaxPooling is done with pool size (3,3) after Batch normalization and ends the block with dropout layer with p = 0.25. The Second Convolution block has convolution2d layer with 64 filters followed by relu activation and batch normalization. The third Convolution block contains convolution2d layers 128 filters with kernel size (3,3) and relu activation and batch Normalization and Maxpooling with pool size (3,3) and dropout as p = 0.25. The fourth convolution block has convolution2d layer with

128 filters, relu activation and batch Normalization.After the 5 convolution blocks the tensors are flattened followed by a fully connected layer of 1024 units and a dropout layer with p = 0.5 and a fully connected layer with number of units equal to number of classes and softmax activation function

5 Results

5.1 Convolutional Neural Network

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)           (None, 178, 178, 32)      896

 max_pooling2d_1 (MaxPooling  (None, 89, 89, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 87, 87, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 43, 43, 64)       0
 2D)

 conv2d_3 (Conv2D)           (None, 41, 41, 128)       73856

 max_pooling2d_3 (MaxPooling  (None, 20, 20, 128)      0
 2D)

 conv2d_4 (Conv2D)           (None, 18, 18, 128)       147584

 max_pooling2d_4 (MaxPooling  (None, 9, 9, 128)        0
 2D)

 conv2d_5 (Conv2D)           (None, 7, 7, 128)         147584

 max_pooling2d_5 (MaxPooling  (None, 3, 3, 128)        0
 2D)

 flatten_1 (Flatten)         (None, 1152)              0

 dropout_1 (Dropout)         (None, 1152)              0

 dense_1 (Dense)             (None, 512)               590336

 dense_2 (Dense)             (None, 4)                 2052

=================================================================
Total params: 980,804
Trainable params: 980,804
Non-trainable params: 0
_________________________________________________________________
Found 1600 images belonging to 4 classes.
Found 400 images belonging to 4 classes.
```

Figure 5: CNN MODEL

Total params: 980,804
Trainable params: 980,804
Non-trainable params: 0
Accuracy: 84

```
Found 400 images belonging to 4 classes.
400
400/400 [==============================] - 21s 52ms/step - loss: 0.4475 - acc: 0.8400

The accuracy of the model is: 0.84 % for loss value 0.45 %.
```

Figure 6: Accuracy

5.1.1 Visualize

We visualized the results. Sometime it might be useful to have a look, what's going on inside our model. For convolutional neural networks we can take a look on particular layers (for example to see what patterns are recognized in each one). In the below, we take a random image, plot it with predicted name, then present what the model sees during 'recognition' on particular level.
We check if model is predicting image and assigning correct class label to it.

Figure 7: predictions

5.1.2 Predictions

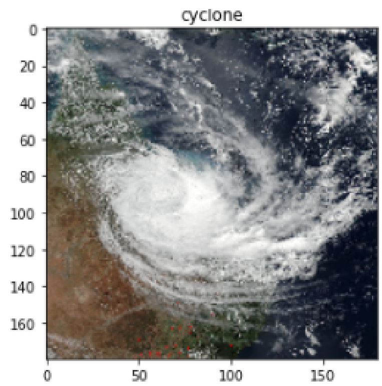


Figure 8: Cyclone

The Predicted label is 0 which corresponds to Cyclone class.

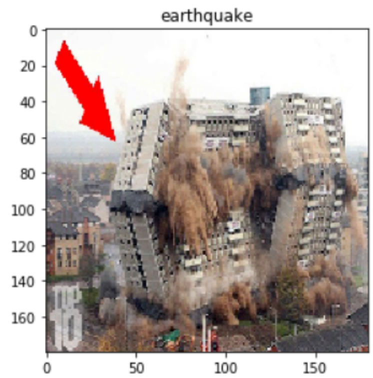


Figure 9: Earthquake

The Predicted label is 1 which corresponds to Earthquake class.

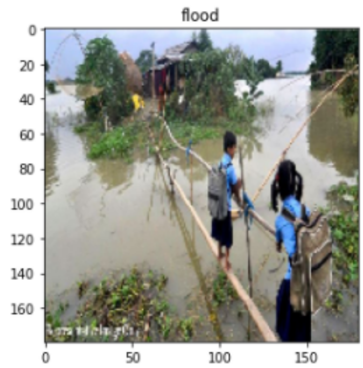


Figure 10: Flood

The Predicted label is 2 which corresponds to Flood class.

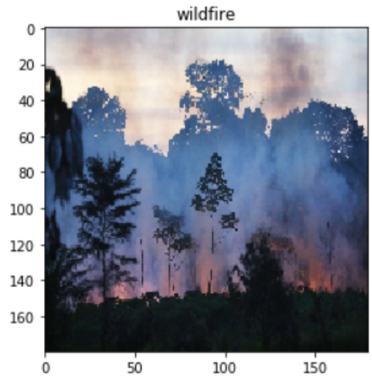


Figure 11: Wildfire

The Predicted label is 3 which corresponds to Wildfire class.

5.2 Application

5.2.1 User Interface

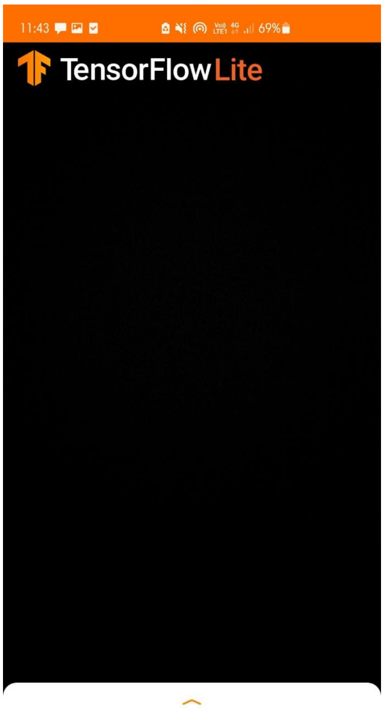


Figure 12: Main Screen of Application

5.2.2 Predictions

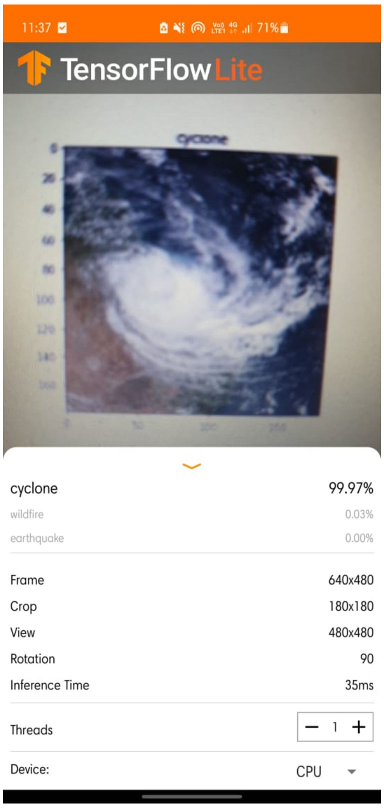


Figure 13: Result

The Predicted class label is 0 which corresponds to cyclone class.

6 Conclusion

With respect to the context of proposed learning model to detect natural disaster, we tried with CNN model and it was trained to:

- To predict the class label of the given image

We explored CNN learning model which is used for natural disaster detection. Earthquakes,floods,wildfires and cyclones are the 4 classes on which proposed method is tested. Therefore, related disasters were taken for identification which spans over these 4 classes. To improve recognition rate in classification process some other pre-trained models and hybrid algorithms can also be used.

7 Future Scope

- The images of wildfires can be monitored using drones. Few more techniques can be used in damage prediction while earthquakes. During cyclone unaffected areas can be monitored and can be used for relief measures.
- With proper reach for the mobile application and considering users upload live images after disasters the area of damage and financial losses and be predicted.
- During floods one can estimate the no.of crops damaged by applying new techniques for the application.

8 References

[1]A UAV-based Forest Fire Detection Algorithm Using Convolutional Neural Network
Publisher: IEEE
Authors:Yanhong Chen; Youmin Zhang; Jing Xin; Yingmin Yi; Ding Liu; Han Liu

[2]Using Machine Learning for Extracting Information from Natural Disaster News Reports
Authors-Alberto Téllez-Valero,Manuel Montes

[3]https://github.com/abhiwalia15/Detecting-Natural-Diasters-with-Keras-and-Deep-Learning-Kaggle/blob/master/train.py