Disaster Images Classification using Prinicipal Component Analysis and Neural Networks

Ioannis Kontogiorgakis[1]

[a]ECE,Technical University of Crete,

Abstract

The objective of the project was to develop an efficient classification model that can accurately classify images into different disaster categories. Three different techniques were explored and evaluated to address this task: Principal Component Analysis (PCA) combined with tabular models (Random Forest and SVM), a custom Convolutional Neural Network (CNN), and Transfer Learning using a pretrained MobileV2-Net model. The experiments and evaluations were performed on a dataset consisting of images from various disaster categories. The results demonstrated that Transfer Learning using the MobileV2-Net model yielded the highest accuracy, showcasing its effectiveness in image classification tasks with limited data.

Keywords: Natural Disasters, Binary Classification, Image Classification, Machine Learning

1. Introduction

Disaster management and response require efficient and timely identification of different disaster events. With the increasing availability of image data, automated image classification systems have gained significant attention for disaster management applications. The objective of this project was to develop a robust and accurate image classification model for identifying various disaster categories, including wildfires, cyclones, earthquakes, and floods. The project aimed to explore different approaches, including traditional machine learning techniques and deep learning models, to compare their performance and determine the most effective method.

The project followed a systematic roadmap to address the image classification problem. The initial approach involved applying PCA to the image dataset to reduce its dimensionality and extract relevant features. These features were then fed into tabular models, specifically Random Forest and Support Vector Machine (SVM), to classify the images into different disaster categories. However, due to the limited amount of available data, the accuracy achieved by these models was not satisfactory.

To overcome the limitations of the tabular models, a custom Convolutional Neural Network (CNN) architecture was designed and implemented. The CNN model was trained on the image dataset to learn the spatial relationships and patterns within the images, enabling more accurate classification. Despite the efforts to optimize the CNN model, the performance remained suboptimal, primarily due to the limited size of the dataset.

Finally, Transfer Learning was employed as a solution to leverage the knowledge learned from a pretrained model for image classification. The MobileV2-Net model, pre-trained on a large-scale dataset, was fine-tuned using the disaster image dataset. This approach showed promising results, as the pretrained model already had learned features relevant to various visual patterns and objects. By leveraging the pre-trained model's knowledge, the transfer learning approach achieved higher accuracy compared to the previous techniques. In this report, we present a detailed description of each approach, including the data preprocessing steps, model architectures, and training procedures. The evaluation metrics, including accuracy, precision, recall, and F1-score, are reported for each technique. Additionally, the challenges encountered, limitations, and possible future directions are discussed. The results highlight the significance of Transfer Learning in addressing the image classification problem with limited data and provide insights for potential applications in disaster management and response systems.

Figure 1: Disaster image dataset.

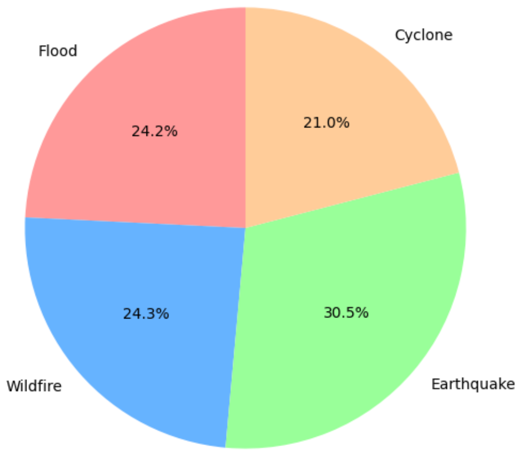

Figure 2: Classes distribution.

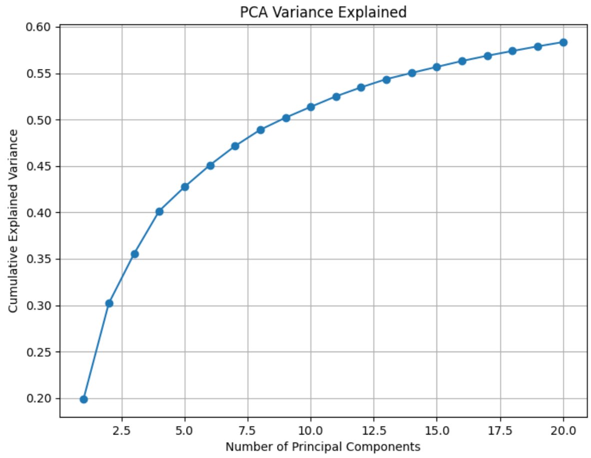

Figure 3: Dataset explained variance.

2. Dataset

The dataset used in this project was acquired from Kaggle and consists of a collection of disaster images and can be aquired from this Link: . The dataset contains a total of 4428 images, which are classified into four different categories:

- Earthquake: 1350 images
- Wildfire: 1077 images
- Flood: 1073 images
- Cyclone: 928 images

Figure 1 showcases a preview of the dataset classes, giving a glimpse of the different disaster categories and the corresponding images. The preview offers a visual representation of the dataset, highlighting the diversity of images within each class.

Figure 2 presents a barplot that illustrates the distribution of images across the different disaster classes. The barplot provides an overview of the dataset's class distribution, showing the number of images available for each disaster category. This information is crucial for understanding the balance or potential class imbalances in the dataset.

This dataset serves as the foundation for training and evaluating various machine learning models in the classification of disaster images. It provides a valuable resource for studying and addressing the challenges associated with automated disaster detection and classification.

3. Methodology

In this section, I outline the methodology employed to tackle the problem of classifying disaster images. The objective of this project is to develop an effective image classification system that can accurately identify different types of disasters based on visual information. This section presents the methodology employed in this study to tackle the task of disaster image classification. The methodology is divided into three experiments, each exploring a different approach. The three experiments conducted are as follows:

3.1. Experiment 1: PCA and Image Classification

In this experiment, we employed Principal Component Analysis (PCA) as a dimensionality reduction technique to preprocess the image data before performing classification. PCA is a widely used method for reducing the dimensionality of high-dimensional data while retaining important information.

According to Bishop's Machine Learning book, PCA involves finding the principal components of the data, which are the directions of maximum variance. These principal components can be obtained through eigendecomposition or singular value decomposition (SVD). The resulting eigenvectors represent the new basis in which the data can be projected onto a lower-dimensional space.

To determine the number of components (k) to select for our PCA analysis, we performed an analysis of the explained variance. The explained variance plot is a valuable tool that illustrates the cumulative amount of variance explained by each principal component. By examining the plot, we can identify the number of components required to retain a significant proportion of the total variance in the data. As we can observe from Figure 3, after $k = 15$, the variance does not change much. For the shake of information loss, we chose $k = 20$ principal components, since according to the variance plot, are sufficient in order to capture a significant amount of variance in the data.

After applying PCA, we visualized the effect of dimensionality reduction by comparing images before and after PCA. Figure 4 showcases a set of sample images from each disaster category before and after PCA. The reduced-dimensional images highlight the preservation of essential features while reducing the dimensionality of the data.

To evaluate the effectiveness of PCA in combination with classification algorithms, we trained Support Vector Machines (SVM) and Random Forest classifiers on the PCA-transformed

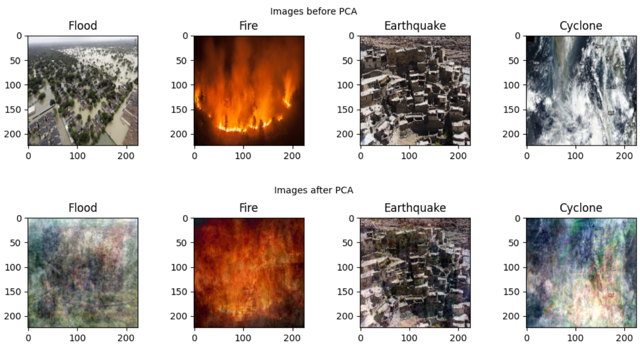

Figure 4: Disaster images before and after PCA.

Algorithm	accuracy	f1-score
PCA + SVM	0.7	0.7
PCA + Random Forest	0.7	0.7

Table 1: Experiment 1 : Evaluation

data. Table 1 presents the accuracy and F1-score obtained from these experiments.

The results of Experiment 1 were neutral, with both the SVM and Random Forest classifiers achieving approximately 0.7 accuracy and F1-score. These modest results suggest that PCA alone may not be sufficient for effectively classifying disaster images. Further experimentation and feature engineering may be necessary to improve the performance of the classification models.

Overall, Experiment 1 served as an initial exploration of using PCA as a preprocessing step for image classification. While the results were not outstanding, they provided valuable insights and set the foundation for subsequent experiments in our methodology.

3.2. Experiment 2: Custom CNN

In the second experiment, we trained a custom Convolutional Neural Network (CNN) model for disaster image classification. CNNs are a powerful class of neural networks specifically designed for processing structured grid-like data, such as images. They are composed of multiple layers, including convolutional, pooling, and fully connected layers, that extract hierarchical features from the input data.

The architecture of our custom CNN model, as depicted in Figure 5, consisted of multiple convolutional and pooling layers followed by fully connected layers. The model aimed to learn spatial hierarchies and extract relevant features from the input images. Specifically, my neural net architecture involves:

- Convolutional Layers: These layers apply convolutional operations to the input image, using learnable filters (kernels) to extract local spatial features. Mathematically, the output of a convolutional layer can be represented as:

$$H_i = \sigma(W_i * H_{i-1} + b_i)$$

where H_i is the output feature map at layer i, W_i is the weight-matrix of the i-th convolutional layer, H_{i-1} is the feature map from the previous layer, b_i is the bias vector and $*$ denotes the convolution operation.

- Pooling Layers: These layers downsample the feature maps, reducing their spatial dimensions while retaining the most salient information. Max pooling is a commonly used technique, where the maximum value within a pooling region is selected as the representative value. The mathematical representation of max pooling is as follows:

$$H_i = max\_pooling(H_{i-1}, pooling\_size)$$

- Fully Connected Layers: These layers connect all neurons from the previous layer to the current layer. Each neuron in the fully connected layer receives inputs from all neurons in the previous layer. Mathematically, the output of a fully connected layer can be calculated as:

$$H_i = \sigma(W_i \cdot H_{i-1} + b_i)$$

where H_i is the output of the fully connected layer, W_i is the weight-matrix, H_{i-1} is theinput from the previous layer, and b_i is the bias vector.

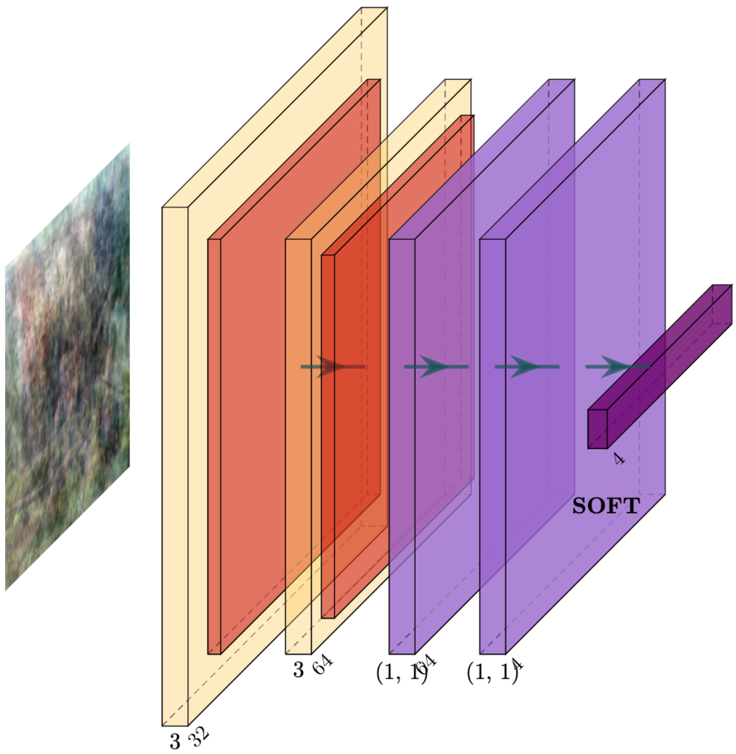

Figure 5: MyCNN neural network architecture.

The input to the CNN model were the resized and preprocessed images from the disaster image dataset. The convolutional layers applied filters to the input, capturing different image features at varying levels of abstraction. The pooling layers reduced the spatial dimensions of the features, preserving the most important information. Finally, the fully connected layers combined the learned features and made predictions based on the extracted representations.

We trained the custom CNN model using the training dataset and evaluated its performance on the test dataset. The accuracy and F1-score of the model are presented in Table 2.

Algorithm	accuracy	f1-score
myCNN	0.60	0.61

Table 2: Experiment 2 : Evaluation

The results of Experiment 2 were not satisfactory, with the custom CNN model achieving an accuracy and F1-score of approximately 0.61. This indicates that the model struggled to effectively classify the disaster images based on the extracted features. Possible reasons for the limited performance could be attributed to the small size of the dataset and the complexity of capturing relevant features from the images. Further optimization and exploration of more advanced architectures and techniques may be necessary to enhance the model's performance.

Experiment 2 provided insights into the challenges of training a custom CNN model for disaster image classification. Although the results were not ideal, they served as a stepping stone for the subsequent experiment, which employed transfer learning to leverage the power of pre-trained models and address the limitations encountered in the previous experiments.

3.3. Experiment 3: Transfer Learning using MobileV2-Net

In this experiment, we aimed to overcome the challenge of having a small amount of data by leveraging the power of transfer learning. Transfer learning is a technique that involves using pre-trained models as a starting point for a new task. By utilizing the knowledge learned from a large-scale dataset, these pre-trained models can be fine-tuned on a smaller target dataset to achieve better performance.

For this experiment, we chose the MobileV2-Net architecture as our base model. MobileV2-Net is a popular deep convolutional neural network architecture that has been pre-trained on a large-scale dataset. It has demonstrated excellent performance in various computer vision tasks.

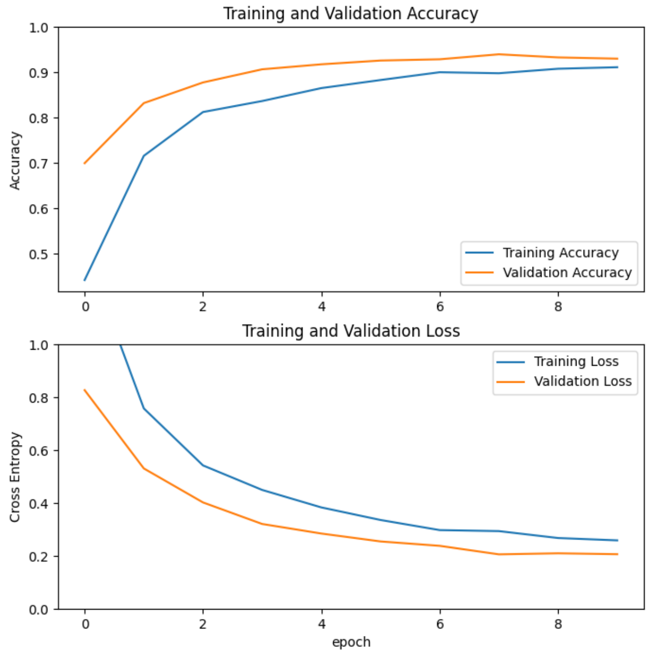

Figure 6: Training and Validation accuracy of MobileV2 Neural Network.

The transfer learning process involves two main steps: freezing the pre-trained layers and adding custom layers for the new task. By freezing the pre-trained layers, we ensure that their learned features are retained and not updated during training. This allows us to focus on training the custom layers that are added on top of the pre-trained model. These custom layers are designed specifically for our disaster image classification task.

During the training process, we observed highly promising results. The model achieved an accuracy of 0.94 as summarized in Table 3, indicating its effectiveness in accurately classifying the disaster images. These results were in line with our expectations, considering the powerful representation capabilities of the MobileV2-Net architecture.

Algorithm	accuracy	f1-score
MobileNet v2	0.94	0.94

Table 3: Experiment 3 : Evaluation

To visualize the training progress, we plotted the train and validation accuracy per epoch, as well as the loss per epoch. Figure 6 showcases these plots, highlighting the increasing accuracy and decreasing loss over the training process. These plots provide valuable insights into the model's learning dynamics and its ability to generalize well on unseen data.

In conclusion, the application of transfer learning using the MobileV2-Net architecture proved to be highly effective in addressing the challenge of limited data. The resulting model achieved remarkable accuracy and f1-score, demonstrating its capability to accurately classify disaster images. This experiment underscores the importance and potential of transfer learning in image classification tasks.

4. Summary and conclusions

In this section, we present and compare the results obtained from the three different experiments conducted in this project: PCA and Image Classification, Custom CNN Model, and Transfer Learning using MobileV2-Net. We also provide a concise conclusion based on the findings. The first experiment involved applying Principal Component Analysis (PCA) to the image dataset and then utilizing the reduced dimensionality data for classification using SVM and Random Forest models. Although the results were not outstanding, with an accuracy and f1-score of approximately 0.7 for both algorithms, it provided a baseline for comparison with the subsequent experiments. In the second experiment, we designed and trained a custom CNN model specifically tailored for the disaster image classification task. The architecture of the model, as depicted in Figure X, included convolutional, pooling, and fully connected layers. However, due to the limited size of the dataset, the model's performance was not optimal, achieving an accuracy and f1-score of approximately 0.61. The third experiment aimed to overcome the data limitations by leveraging transfer learning. We utilized the MobileV2-Net architecture, a pre-trained model, and fine-tuned it on our disaster image dataset. The results were highly promising, with an accuracy of 0.94 and an f1-score of 0.94. This demonstrates the power of transfer learning in leveraging pre-trained models to achieve exceptional performance, even with limited data.

Comparing the three experiments, we observed that the transfer learning approach outperformed the other two methods. The combination of a powerful pre-trained model and fine-tuning on the target dataset proved to be highly effective in capturing relevant features for accurate classification. In contrast, the other experiments struggled to achieve satisfactory results due to the limited size of the dataset.

In conclusion, this project highlights the importance of appropriate methodologies and techniques for disaster image classification. While PCA and custom CNN models provide viable approaches, transfer learning with a pre-trained model demonstrated superior performance. The successful implementation of transfer learning using the MobileV2-Net architecture showcases its potential in addressing the challenges posed by limited data. These findings contribute to the advancement of image classification in disaster scenarios, paving the way for more accurate and efficient detection and response systems.