Natural Disaster Classifier Report

Jared Paul Guevara, Laura Jones, Brandon Robello

April 2024

1 Abstract

The following report summarizes work on a model that predicts the type and extent of damage using a set of satellite images of the aftermath of Hurricane Matthew, Southern California fires, and Midwest flooding. The image data is examined as matrices, with additional features made to explore color and texture features as well as value distributions across the dataset.

For Task A, we built a classifier to distinguish images between Midwest flooding and So-cal fires. Our team identified ideal feature combinations as well as model parameters and reached a cross validation score of .985 across 10 folds of the data. In Task B, the goal was to correctly label disasters on a 4-point scale of damage extent (0-3), with the integer increasing with the intensity of the damage.

The project was helpful in learning to create a successful image classifier without the use of a convolutional neural network backed by rigorous exploratory data analysis and feature transformation.

2 Introduction

This project and dataset stems from an article implementing a model to assess building damage using aerial imagery [GHS19]. This article outlines the collection of the dataset, description of the dataset, analysis using a pretrained ResNet50 model, and the evaluation of the model. Much of the focus was in evaluating building damage, which emphasized the polygons representing structures in the image. This method was not integrated in our analysis.

Convolutional neural networks (CNNs) are the current highest development in understanding images from an algorithm featurespace. The following analysis is more deliberate, and extracts features from the dataset directly for the training of the model. This examination, while not the forefront in image processing research, is an exercise that improved our understanding of the intricacies of computer vision and exemplifies that more traditional machine learning techniques are still effective at solving complex problems.

3 Description of Data

3.1 General Characteristics

For a total of 26,535 images, there were 11,151 images of Hurricane Matthew, 8380 images of Southern California Fires, and 7,004 images of Midwest flooding. Initial exploratory data analysis (EDA) included size distributions (see supplemental materials pg. 7) as well as color distributions among the damage types (see supplemental materials pg. 8). The images in the hurricane Matthew, Midwest flooding, and Socal fire subsets had average aspect ratios of 1:1, 2:1, and 2:1, respectively. In addition, the hurricane and fire datasets had similar color channel intensities (about .42 for blue, .44 for green, and .35 for red); the flooding images had differing values at .34 for blue, .38 for green, and .28 for red. This difference likely made it easier to categorize images as either flooding or fire instances for Task A.

For Task B, it was important to visualize the distribution of damage labels across the three types of disasters, as seen in figure 1. An interesting hurdle was label imbalance; successful classification of Hurricane Matthew images was key for model success, so Imbalanced-Learn was used to make the data set generalized with a more equal ratio of classes (see Methods on pg. 3). For Task A, the imbalance was less significant, as the goal did not heavily involve damage labels.

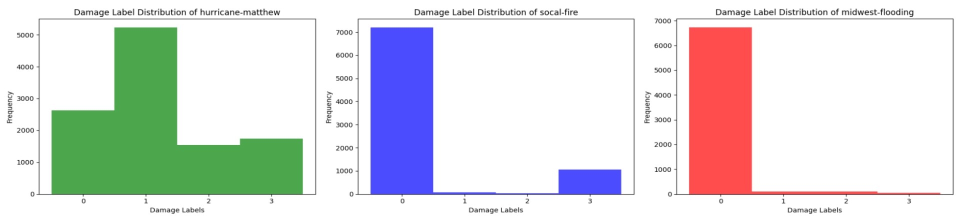

Figure 1: Distribution of damage labels across 3 disaster types

GroupBy operations were applied to find potential features of interest among the Socal fire, Midwest flooding, and hurricane Matthew dataframes. From this analysis, it was concluded that the socal fire images had more pixels than Midwest flooding, as well as a higher pixel intensity on average. Examination of trends in the Hurricane Matthew images revealed that there were about half as many pixels on average for images with a damage label '3' than there were for the other three damage labels (0,1,and 2), but the pixel intensity was fairly consistent across all four classes (see summary in supplemental materials, pg. 9).

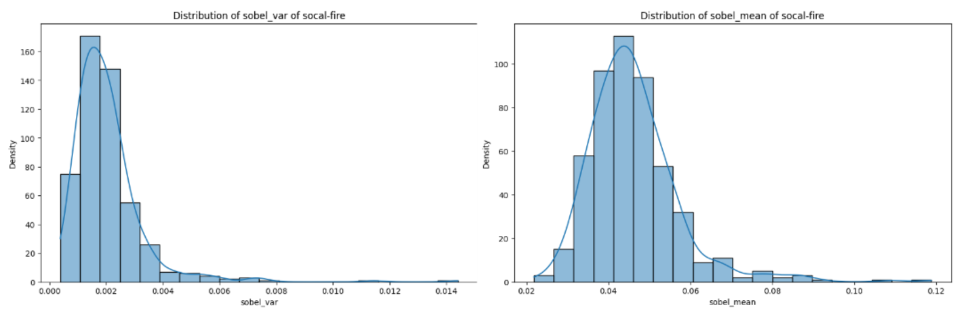

Figure 2: Distribution of Sobel Mean and Variance

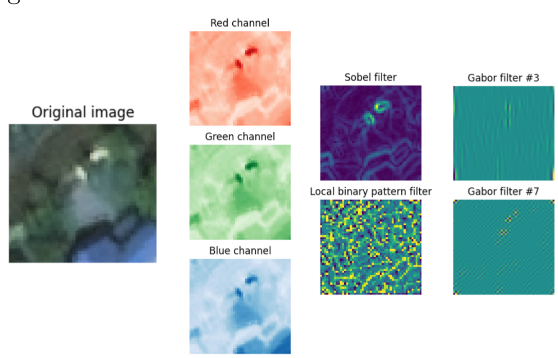

Figure 3: Kernel Filter Examples

3.2 Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique that reduces large datasets into smaller ones that still roughly maintain the information from the original set [Jaa24]. Accuracy may be traded for faster and easier processing of the adjusted dataset. In addition, PCA can help with generalization and balancing the bias variance tradeoff by eliminating some components of the data set that lack variance. It effectively regularizes the dataset and prevents overfitting of the model, potentially improving performance when inferring unseen data.

PCA was applied using Sklearn's decomposition module, after scaling the image matrices, disaster types, and labels. Initially, the analysis technique revealed little in the way of differentiating between Socal fire and Midwest flooding disasters, with the principal components intermingling with one another (Fig. 4).

3.3 Feature Extraction

Some features were gathered from the original matrices using the Sobel, Gabor, and Local Binary Pattern (LBP) filters. The Sobel filter is an important kernel for edge detection as it "emphasizes regions of high spatial frequency that correspond to edges"[Fis+03]. Gabor and LBP filters are often used for texture analysis as model features [Tra98; Pie10], which is relevant to this project for differentiating between structures and topography in our image dataset. Later exploration of feature importance using Sklearn's classification report (see Methods section, pg. 4) indicated that LBP variance, LBP mean, and Sobel mean had the most impact in the model's classification between fire and flooding images (pg. 10).

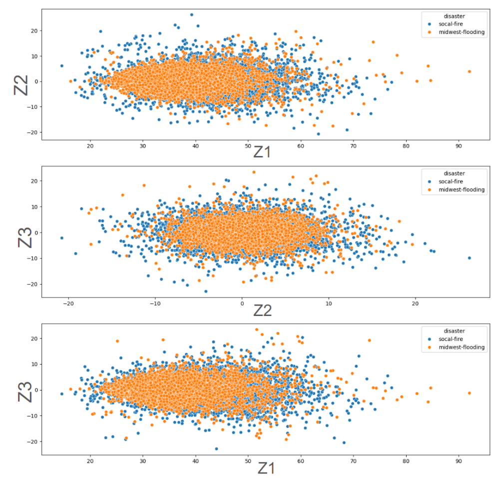

Figure 4: Initial PCA to find component clustering among disaster types

One transformed feature appended to the dataframes was the ratio of pixels that met a certain intensity value over the area of the image, with a new column being calculated for each color channel (fig.5). An interesting trend was that the ratio of blue pixels meeting the intensity threshold of 150 decreased as the damage label increased for the Socal fires dataset. It also seemed that the ratio of the area decreased as the severity of the damage increased.

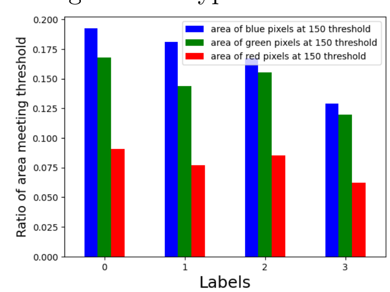

Figure 5: Comparison of color channels meeting threshold

4 Methods

4.1 Features

The data loading, EDA, and model generation was completed on Google Collab and JupyterNotebook. The features included to begin generating models to complete Task A are summarized below:

disaster	
label	height
width	aspect_ratio
total_pixels	avg_pixel_int
lbp_var	lbp_mean
sobel_mean	gabor_7_var
gabor_3_var	area of red pixels at 150 threshold
area of green pixels at 150 threshold	area of blue pixels at 150 threshold

The number of features was reduced from the larger sample of gabor filter values by generating a RandomForestClassifier and extracting the feature importance, a value that identifies what features of the data contribute most to the most accurate predictions. Figures 11 and 12 in the supplemental materials (pg. 10) compares the feature importance for Task A and Task B, which aided in selecting the features used for Principal Component Analysis (PCA).

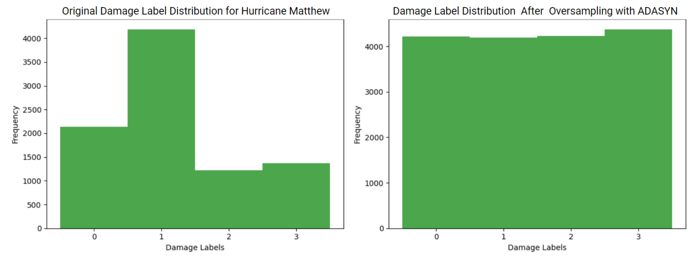

Figure 6: Comparison of original and oversampled label distribution using ImbLearn

An important step centered around resampling to circumvent damage label imbalance. As seen in figure 6, final damage label distribution is nearly equalized using ADASYN, an oversampling technique that "generates synthetic samples in areas where classification is difficult" [M23]. Other remedies included RUS (random under-sampling) and ROS (random over-sampling), and the results of the different methods are reviewed on pg. 5 in the Summary of Results section.

4.2 Models

To accomplish Task A, LinearSVC and RandomForestClassifier models were trained and tested to differentiate between the Midwest flooding and Socal fire disaster images. The selected models are strong classifiers, and by using the default parameters, we were able to attain > 0.98 accuracy scores (see summary of results on pg. 5).

The goal of Task B was to correctly identify the damage label of an image in the Hurricane Matthew dataset. LinearSVC and the RandomForestClassifier was implemented for this Task as well. The evaluation metric was an F1 score, which strives to diminish false negatives and false positives by taking a harmonic mean of precision and recall[Sha23].

The modeling pipeline for both tasks involved splitting the data into training and testing, fitting the data, and generating an accuracy score. Sklearn's SVM, ensemble, and pipeline libraries were implemented to generate and evaluate the models, while model selection and standard scaler were used to process and split the data. GridSearchCV was also implemented to identify best parameters for both LinearSVC and RandomForestClassifier model variations. For LinearSVC, only the regularization parameter 'C' and the number of iterations was identified, while number of estimators, max features, and tree depth could be tuned for Random Forest. In addition, impactful features were highlighted using Sklearn's classification report, which returned accuracy, precision, and recall evaluation metrics. This was a key step in identifying the values that model accuracy depended on.

5 Summary of Results

5.1 Task A

Initial training scores for differentiating between fire and flooding damage using LinearSVC and RandomForestClassifier were 0.982 and 0.995, with average cross-validation across 10 folds being 0.982 and .985, respectively. Data was accumulated into confusion matrices for each model and their training and validation scores (see supplemental materials pg. 11); a confusion matrix summarizes the true positives and true negatives (which should be maximized) and the false positives and false negatives (which should be minimized). The RandomForestClassifier outperformed the LinearSVC model on both the training and validation sets on all metrics, as seen in figure 7.

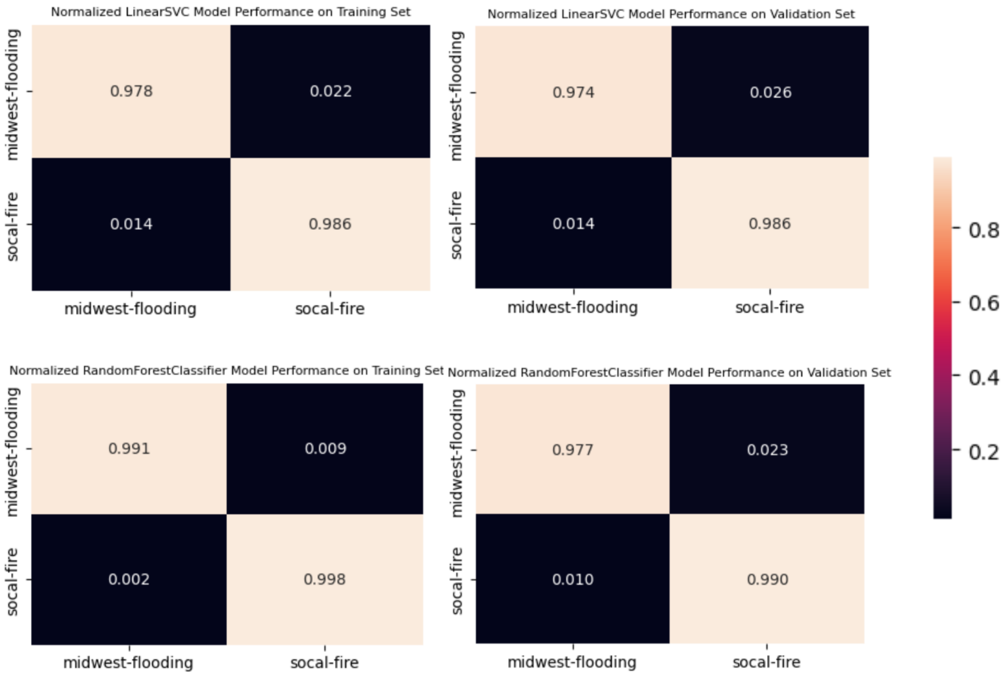

Figure 7: Normalized confusion matrix results for Task A

Random forest may have outperformed LinearSVC because of the finely tuned decision making process, an outcome of having many and deep layers of decision trees. In contrast, LinearSVC operates by finding a classification hyperplane that best defines the separation of data points.

5.2 Task B

Training scores for Linear SVC and Random Forest for classifying hurricane damage labels were 0.549 and 0.749. Average cross-validation scores were 0.550 and 0.622. The weighted average of F1 scores, which are more insightful for imbalanced datasets, grew from 0.515, to 0.548, to 0.564 after RUS, ROS, and ADASYN label balancing.

Techniques	**F1 Accuracy**	**0 Labels**	**1 Labels**	**2 Labels**	**3 Labels**
Original	n/a	2138	4190	1217	1375
RUS	0.515	1217	1217	1217	1217
ROS	0.548	4190	4190	4190	4190
ADASYN	0.564	4214	4190	4226	4377

The above figure shows that the ADASYN resampling technique yielded the highest accuracy prior to applying the optimum pipeline while also maintaining precision for the majority class (label 1). These results suggest better overall learning despite the decrease in accuracy.

6 Discussion

6.1 Model Appraisal

Task A proved to be a success, with the most constitutive adjustments to the model arriving from the identification of ideal features and hyperparameters using GridSearchCV and a function identifying the best features. LBP and Sobel

Task B was saw lower F1 scores as the task to identify damage labels proved challenging. Balancing the damage labels faintly improved the evaluation scores, and the most important features tended towards LBP and Gabor filter metrics that highlighted textural differences in the images.

6.2 Project Extensions

Additional features as well as feature reduction (see 'PCA', pg. 2) could potentially increase model accuracy. A convolutional neural network, such as a Residual Network (ResNet), has the potential to raise image classification success, but would require greater resources to evaluate using the extensive dataset.

References

[Tra98] Trapp. *Active Stereo Vision*. Nov. 1998. URL: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TRAPP1/filter.html (visited on 04/22/2024).

[Fis+03] Fisher et al. *Feature Detectors - Sobel Edge Detector*. 2003. URL: https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm.

[Pie10] Matti Pietikäinen. "Local binary patterns". In: *Scholarpedia journal* 5.3 (Jan. 1, 2010), p. 9775. DOI: 10.4249/scholarpedia.9775. URL: http://www.scholarpedia.org/article/Local_Binary_Patterns.

[GHS19] Ritwik Gupta, Richard Hosfelt, and Sandra Sajeev. "xBD: A Dataset for Assessing Building Damage from Satellite Imagery". In: *arXiv* (Nov. 21, 2019).

[M23] Vijay M. "Data Imbalance: How is ADASYN different from SMOTE?" In: (Sept. 26, 2023). URL: https://medium.com/@penpencil.blr/data-imbalance-how-is-adasyn-different-from-smote-f4eba54867ab#:~:text=SMOTE%20generates%20synthetic%20samples%20uniformly,on%20the%20dataset's%20specific%20challenges..

[Sha23] Natasha Sharma. *Understanding and Applying F1 Score: AI Evaluation Essentials with Hands-On Coding Example*. June 6, 2023. URL: https://arize.com/blog-course/f1-score/#:~:text=F1%20score%20is%20a%20measure,can%20be%20modified%20into%20F0..

[Jaa24] Zakaria Jaadi. *A Step-by-Step Explanation of Principal Component Analysis (PCA)*. Feb. 23, 2024. URL: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

7 Supplemental Materials

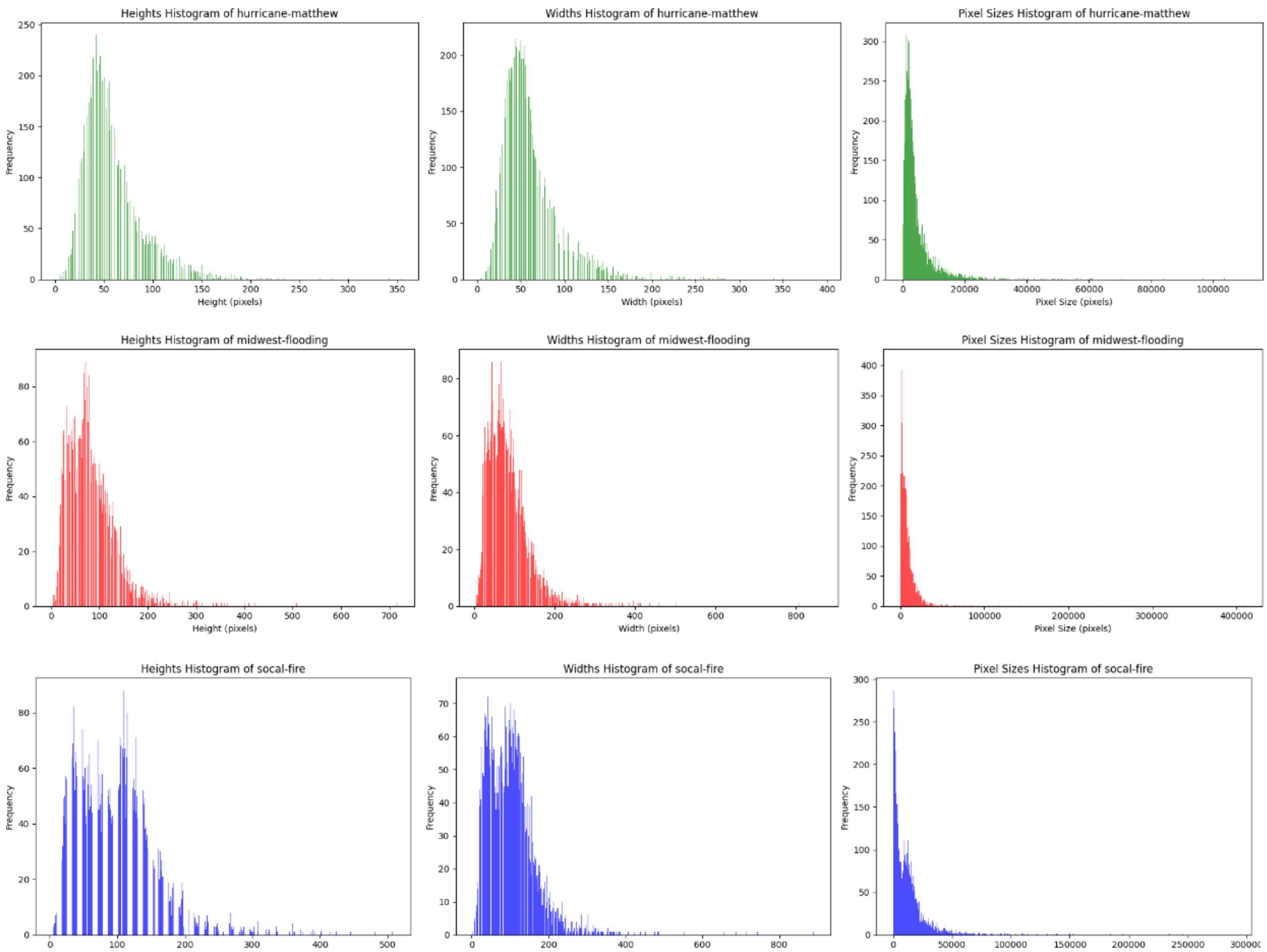

Figure 8: Distribution of height, width, and pixel size across 3 disaster types

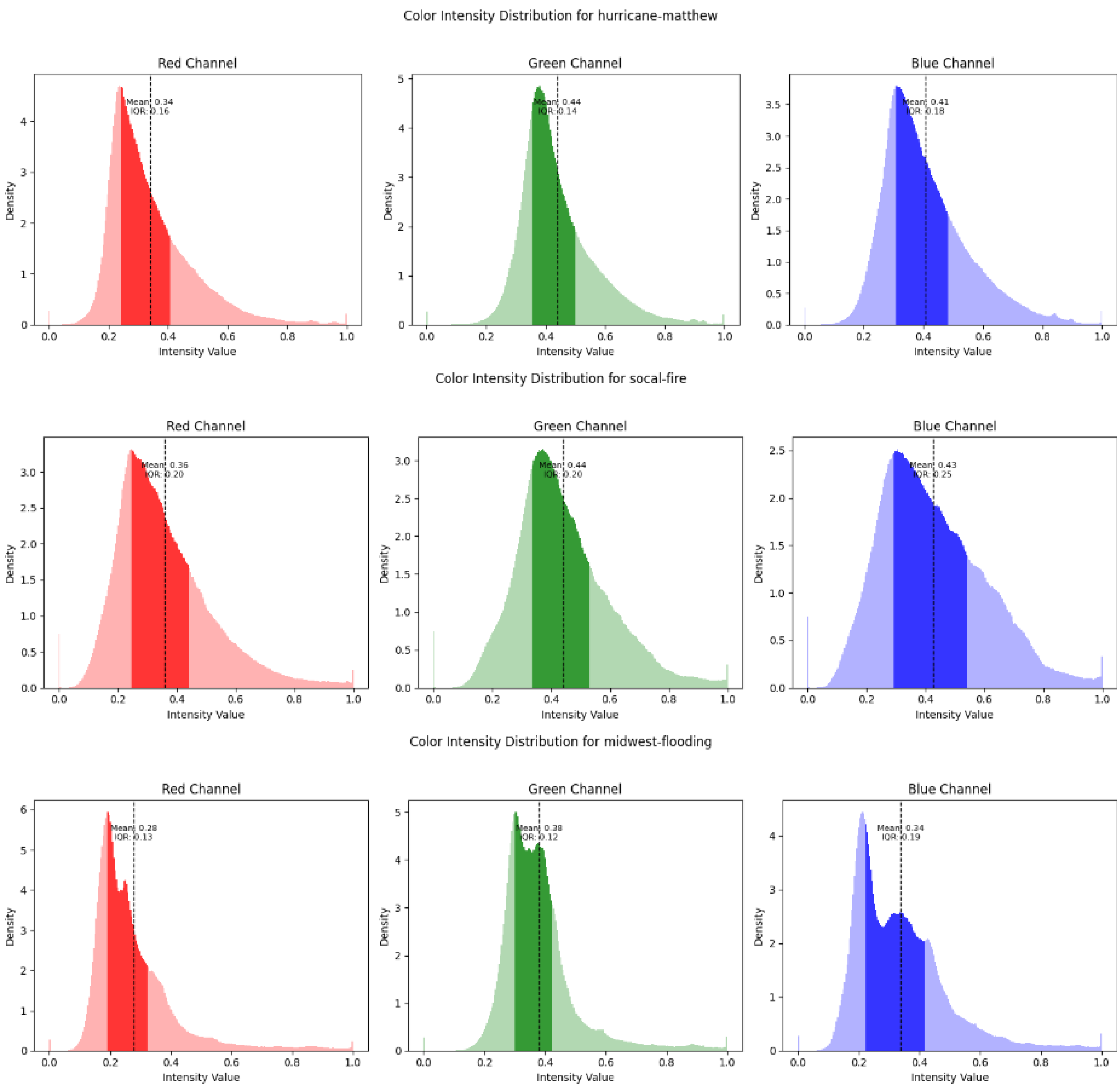

Figure 9: Distribution of color channels across 3 disaster types

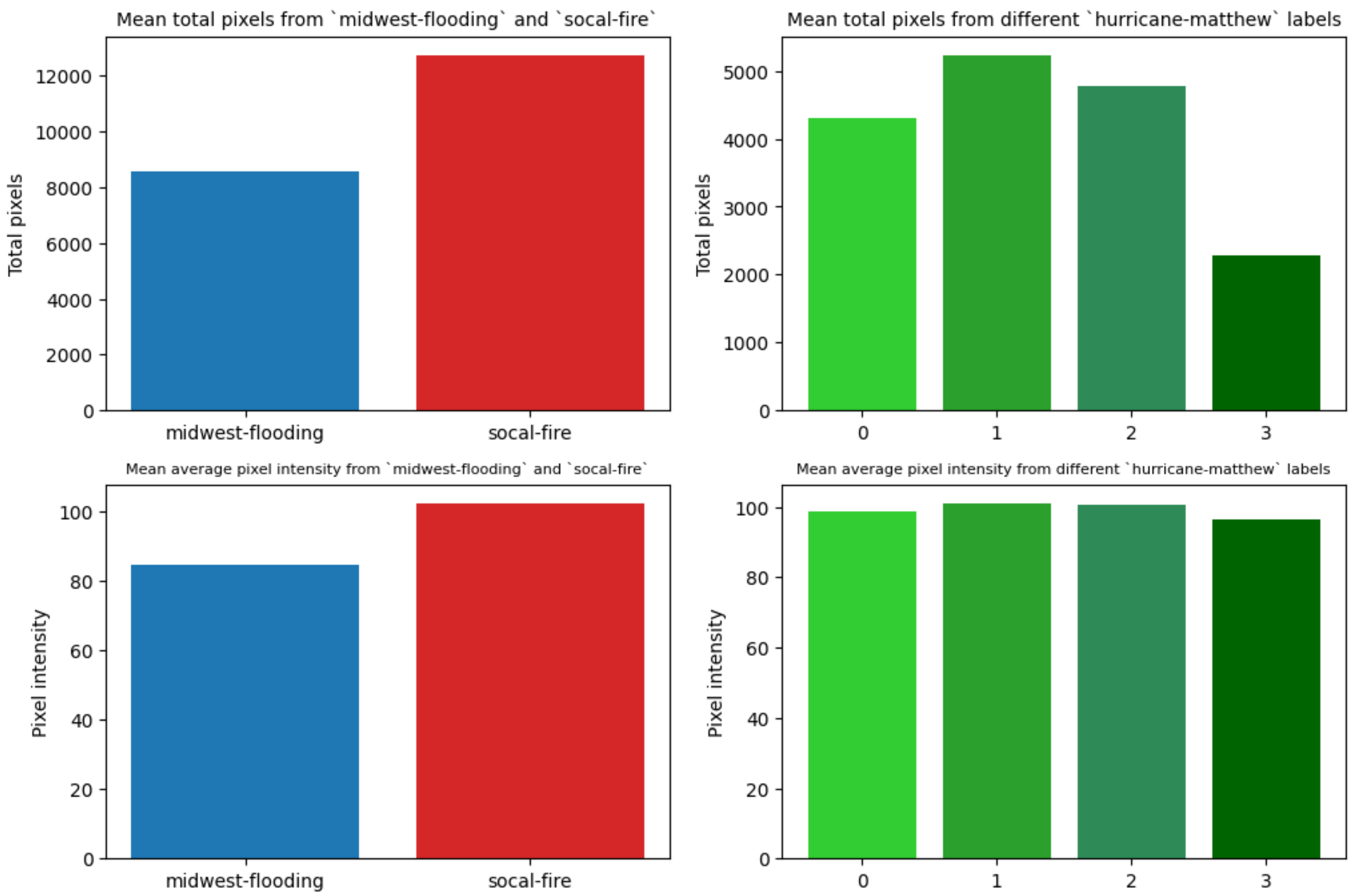

Figure 10: Results of GroupBy operations across three datasets

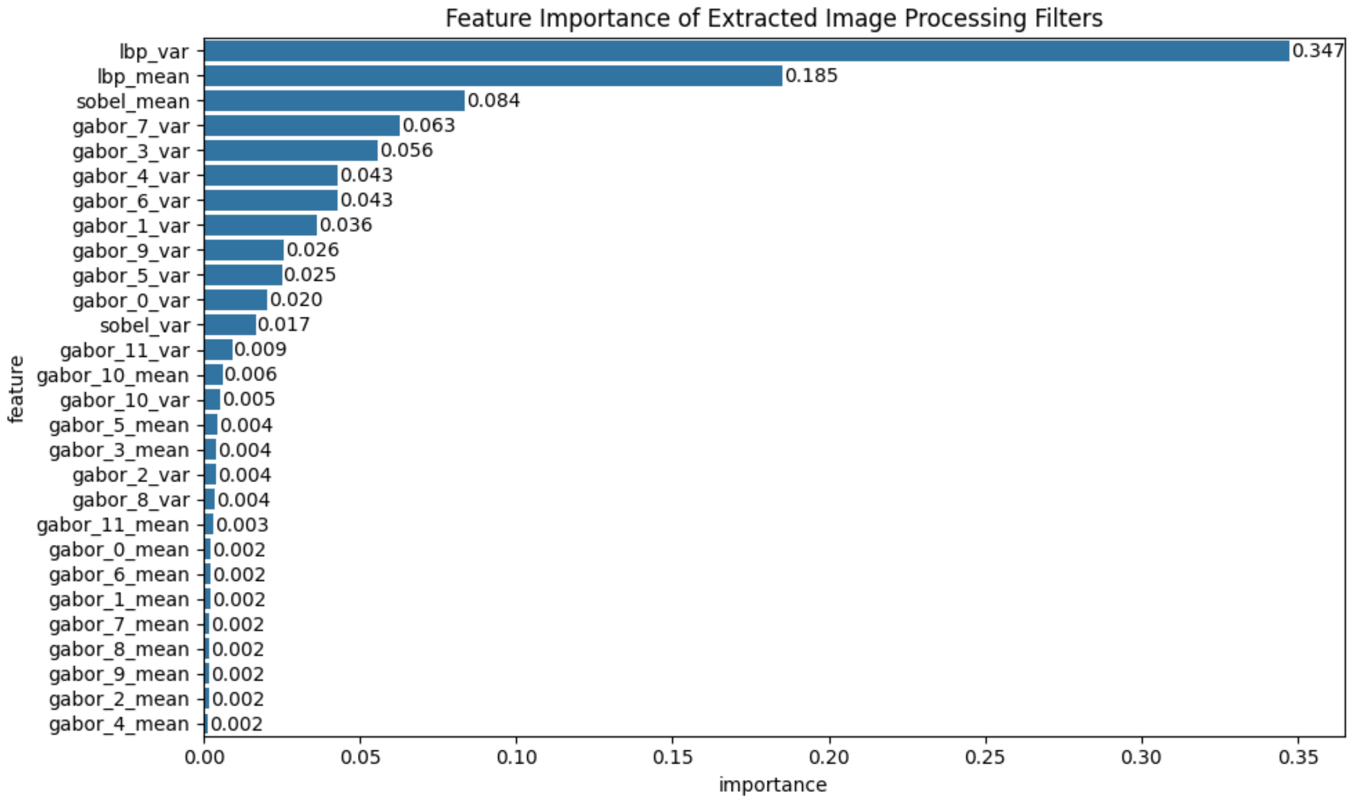

Figure 11: Feature importance for Task A

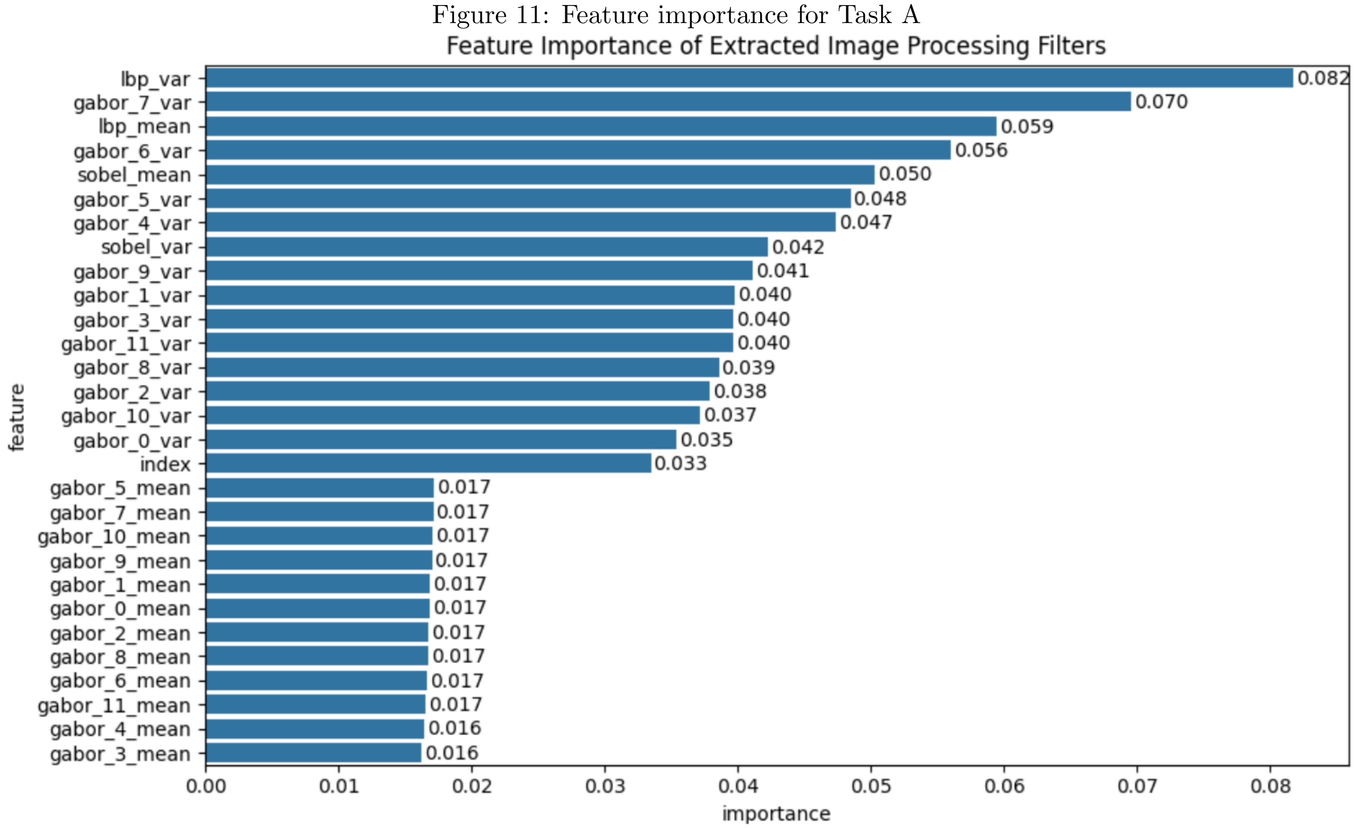

Figure 12: Feature importance for Task B

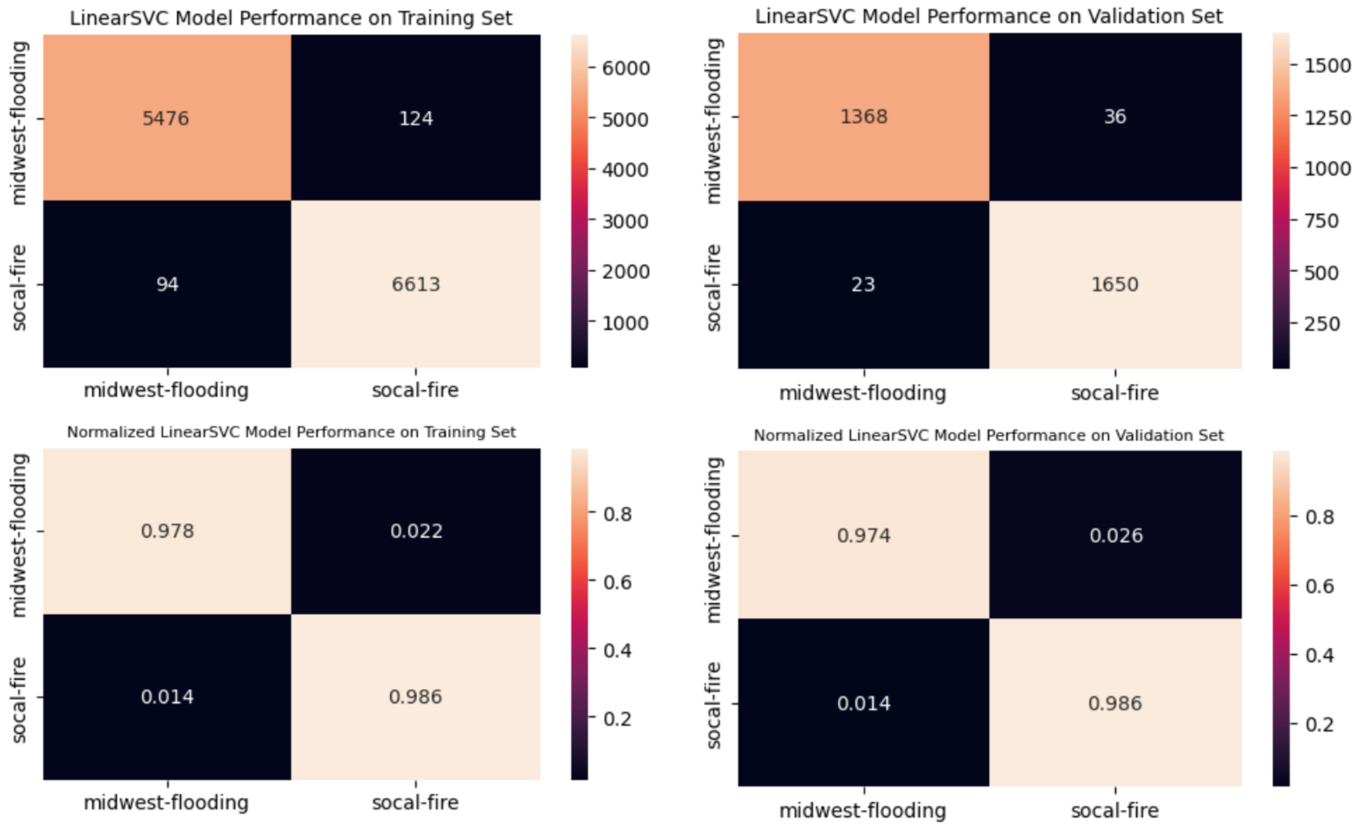

Figure 13: Confusion matrix for Task A, LinearSVC

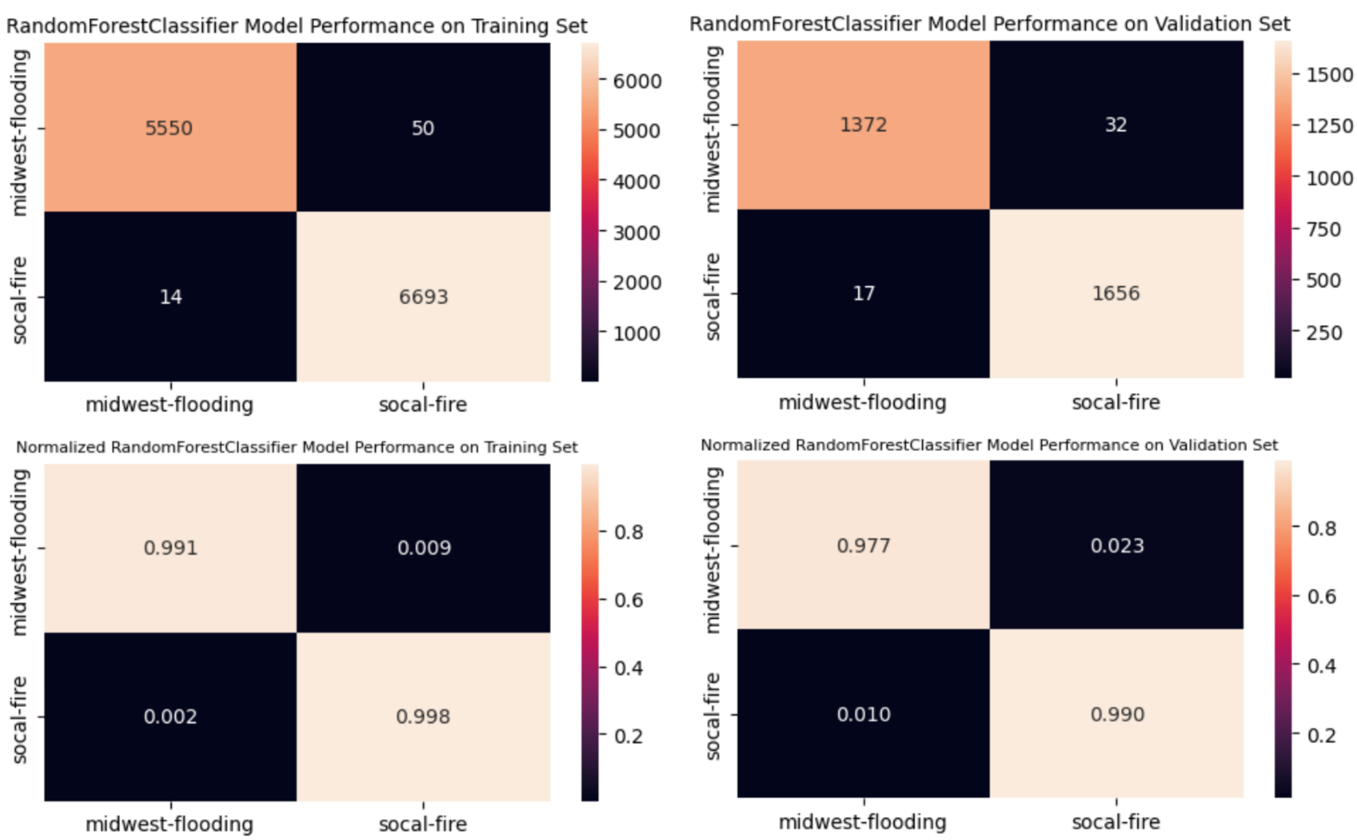

Figure 14: Confusion matrix for Task A, Random Forest Classifier