Robust Disaster Damage Assessment: Leveraging Large Pretrained Models

Raashi Mohan

May 2024

Carnegie Mellon University
Computer Science Department
School of Computer Science
Pittsburgh, PA, 15213

Advisors

Aditi Raghunathan
Amrith Setlur, Saurabh Garg

Abstract

Robustness to distribution shifts is essential for utilizing machine learning models in real-world applications. Nevertheless, existing techniques that enhance the performance of machine learning in the presence of these shifts have primarily focused on shifts that are well-defined and uncomplicated — no method has proven successful in improving performance in the open-ended scenario considered in this study. Unfortunately, the de-facto standard of simply using existing data from previous related events to create models tailored towards specific tasks often falls short in several real-world situations, as these types of strategies are designed for conventional machine learning benchmarks, where a wealth of labels is available, and distribution shifts are absent. In light of these challenges, this thesis aims to devise novel approaches for effectively performing disaster assessment after natural disasters, while leveraging OpenAI's CLIP as a large pre-trained zero-shot backbone, with the goal of enhancing performance in the presence of distribution shifts.

Contents

Chapter 1

Introduction

Natural and artificial disasters affect millions yearly and cost billions in economic damage. In 2017, economic losses due to weather disasters were estimated at $306.2 billion in the US alone. Worldwide, 20 million people are displaced annually due to extreme weather events [1]. Compounded by climate change, many experts expect that recurrent disasters such as wildfires, floods, and tropical cyclones will only become more prevalent in the coming decades. Damage assessment, the process of quantifying the effects of disasters to equip crisis responders with situational awareness and help them plan rescue and recovery efforts, is currently performed manually by teams of experts, and can take weeks to months to complete. This research is in collaboration with the SKAI team at Google Research, which been attempting to address this problem by utilizing machine learning, computer vision, and remote sensing to improve damage assessment efficiency to take hours instead of months. However, models created thus far still struggle to generalize to unseen disasters.

Machine learning typically requires a large amount of labeled data and performs best on inputs that come from the same distribution as the training data. Unfortunately, manually labeling additional data is time-consuming, expensive, and error-prone, and in this type of application, it is essential that models are able to generalize to new and unseen disasters. Furthermore, there is noise inherent to remote sensing data (i.e. angle of imagery captured, time of day, change in geography or landscape). The goal of the this research is to develop methods that address these challenges and enable researchers to make generalizable predictions on remote sensing data.

Chapter 2

Background

2.1 Distribution Shift

In the context of machine learning, particularly for tasks involving image recognition, achieving good performance hinges on the quality and representativeness of the training data. However, a common challenge arises when a model trained on a specific data distribution encounters real-world data that deviates from that distribution – a phenomenon referred to as distribution shift. For example, temporal shift arises from changes in data characteristics over time, attributable to evolving external factors, societal trends, or alterations in data collection methodologies. Models trained on historical data may struggle to adapt to emergent patterns or behaviors. Furthermore, cultural or geographical variations in data introduce another dimension of distribution shift. Data collected from differing geographical regions or cultural milieus may exhibit variations in language, customs, or social norms. Consequently, models trained on data from one region or culture may falter when generalized to others due to these distinctions [2].

Utilizing zero-shot learning models presents an innovative approach to mitigating distribution shift in machine learning applications. Zero-shot learning refers to the ability of models to generalize to unseen classes or domains without explicit training data, relying instead on transferable knowledge encoded during initial training [3]. These models inherently possess a degree of robustness to distribution shift due to their ability to generalize beyond the training data distribution. By leveraging semantic similarities and transferable representations, these models can make predictions on data distributions that differ from those encountered during training, effectively mitigating the impact of distribution shift.

2.2 CLIP: Contrastive Language-Image Pretraining

CLIP (Contrastive Language-Image Pretraining), a deep learning model developed by OpenAI, can understand images and text jointly. CLIP aims to learn a shared embedding space for images and text, enabling the model to understand relationships and semantics between images and their associated textual descriptions [4]. More specifically, CLIP takes in an image and a series of text prompts and returns similarity scores between the image and text encodings. This score indicates how well the text description matches the image content. Ultimately, CLIP's embedding space is rich and high-dimensional, enabling nuanced representations of complex relationships between images and text, making it an ideal backbone for these image-related experiments.

Leveraging CLIP as a pre-trained model allows for transfer learning. CLIP is pre-trained on a diverse range of internet data, encompassing a vast array of visual and textual concepts. This pretraining enables the model to capture high-level semantics, features, and patterns from a broad spectrum of images and associated text [4]. Utilizing and fine-tuning CLIP on specific datasets, like the xBD dataset mentioned above, can enhance its understanding of domain-specific relationships without starting from scratch. Ultimately, CLIP has demonstrated competitive performance across various benchmarks, indicating its effectiveness in understanding and reasoning about images and text. Its success in tasks like zero-shot classification, image-text retrieval, and visual question answering showcases its capabilities in multimodal understanding.

At its core, CLIP utilizes a separate image encoder and text encoder to process image and text inputs. The image encoder breaks down input images into parts and encodes them, while the text encoder turns given text prompts into numerical representations. The text and image encoder are trained by bringing their encodings closer together when an image and its given caption match (and vice versa for an image and a different caption). Through this contrastive training on many examples, CLIP builds a common learned space where related images and their text descriptions reside close together. Ultimately, this allows CLIP to find images that match a text description by comparing their positions in the common space.

2.2.1 CLIP Image Encoder

One image encoder that CLIP utilizes is a Vision Transformer (ViT) model. Unlike traditional Convolutional Neural Networks (CNNs) that process images pixel-by-pixel, ViTs take a different approach. The ViT first splits the image into smaller squares or rectangles called patches. Each image patch is then flattened and fed through a series of transformer layers. This allows the model to focus on smaller regions of the image and learn the relationships between them [5]. The original CLIP implementation used a ViT-B/16 architecture, a specific configuration with 16 encoder layers and a patch size of 16 pixels. Throughout this research, I utilize this powerful image encoder by itself, separately from CLIP's text encoder.

2.3 The xBD Dataset

The xBD dataset stands as a pivotal resource for the experiments conducted within this thesis, providing a comprehensive repository of satellite imagery encompassing diverse disaster scenarios. While many existing satellite imagery datasets are confined to singular disaster types, the xBD dataset distinguishes itself by offering over 45,000 km^2 of meticulously labeled pre- and post-disaster imagery, as is demonstrated in Figure 2.1 [6]. These images are annotated with damage classification labels, including 'No Damage', 'Minor Damage', 'Major Damage', and 'Destroyed'.

One of the distinguishing features of the xBD dataset, in comparison to other datasets, is its breadth of coverage across different types of disasters. Spanning events such as hurricanes, tornadoes, wildfires, earthquakes, and tsunamis, among others, this dataset encapsulates the multifaceted nature of natural calamities. By encompassing such a wide array of disaster types, the xBD dataset enables researchers to develop and evaluate models that are robust and adaptable across various environmental crises.

Figure 2.1: Example paired images from xBD dataset [6]. Pre-disaster imagery (top) and post-disaster imagery (bottom) for Hurricane Harvey, Joplin tornado, Lower Puna volcanic eruption, and Sunda Strait tsunami

2.3.1 Limitations of xBD

While the xBD dataset offers a rich and diverse collection of satellite imagery for disaster analysis, there is one prominent limitation to this dataset – the imbalance between positive and negative examples. Specifically, there are significantly more negative examples (instances where no damage to the environment is observed) compared to positive examples (where disasters have inflicted damage).

This class imbalance can pose challenges during model training and evaluation. Machine learning algorithms trained on imbalanced datasets may exhibit a bias towards the majority class (in this case, negative examples), potentially leading to suboptimal performance in identifying and accurately classifying instances of environmental damage. Furthermore, the scarcity of positive examples relative to negative examples may hinder the ability of models to learn meaningful patterns and features associated with damaged areas, thereby limiting their effectiveness in real-world applications.

Chapter 3

Experiments and Results

3.1 Notes

The primary objective of these experiments is to develop a robust binary classifier capable of accurately distinguishing images into two classes: positive, representing areas that have incurred damage, and negative, indicating areas that remain undamaged after a natural disaster has occurred. The resulting classifier should perform well on new and unseen datasets as well so that it can be used to facilitate effective disaster assessment and response efforts.

3.1.1 Criteria for Evaluation

To evaluate the performance of the classifiers, I employ two key criteria: accuracy and robustness. Accuracy measures the model's ability to correctly classify images within the training distribution (often referred to as Tier 1 data), where the data distribution aligns with that of the training set. However, to ensure the practical utility of our classifier in real-world scenarios, I also assess its performance on out-of-distribution or unseen data (often referred to as Tier 3 data). Robustness, in this context, refers to the model's capability to generalize well to unseen data and maintain high performance even when faced with data distributions different from those encountered during training.

My ultimate aim is to develop a binary classifier that not only achieves high accuracy on the training distribution but also demonstrates robustness by performing effectively on unseen data. By prioritizing both accuracy and robustness, I endeavor to create a classifier that can reliably identify areas affected by disasters across various scenarios and environmental conditions.

3.2 Evaluating CLIP Off-The-Shelf

The first major experiment focused on evaluating the performance of the CLIP model on our selected dataset (without any additional fine-tuning). Despite its potential, CLIP exhibited limitations in discriminating between pre- and post-disaster images, as well as in distinguishing between damaged and undamaged aerial images. My initial focus was on prompt engineering, aimed at optimizing text prompts to enhance CLIP's performance. I selected several classes for evaluation, including descriptors like the following:

1. "disaster-stricken environment" vs. "baseline environment"
2. "satellite image of an area affected by a natural disaster" vs. "satellite image of an area"

3. “This is a satellite image of an area affected by a natural disaster” vs. “This is a satellite image of an area”.
4. “This is a satellite image of a damaged area” vs. “This is a satellite image of an undamaged area.”
5. “This is a satellite image of an area affected by a <disaster>” vs. “This is a satellite image of an area”

However, our findings revealed that CLIP struggled to differentiate effectively, consistently returning a near 50/50 probability distribution for all input images, irrespective of their disaster status. This trend persisted across approximately 18,000 post-disaster images, as depicted in Figure 3.1.

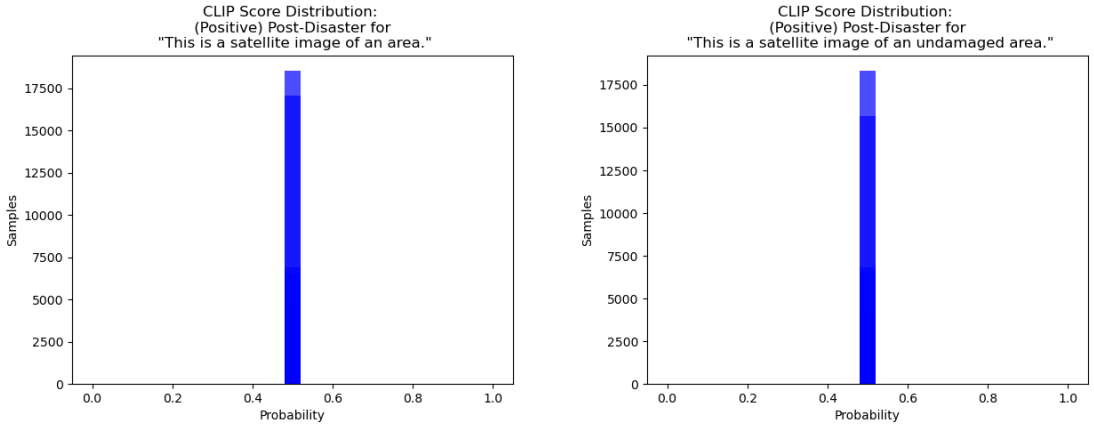

Figure 3.1: Histogram of probabilities for classes “This is a satellite image of an area affected by a natural disaster” and “This is a satellite image of an area” on all post-disaster images.

Further analysis of CLIP’s class scores prior to normalization provided insights into potential avenues for improvement. Although subtle differences were observed, particularly for more descriptive prompts, these variations were insufficient to achieve meaningful discrimination, as shown in Figure 3.2. This trend is also captured by the resulting confusion matrices in Figure 3.3, which demonstrate the inability of CLIP to differentiate between both types of images.

3.2.1 A Deeper Look Into CLIP Scoring

TODO: CLIP OTS scores (11/14 images)

3.3 Binary Classification Through Linear Probing

In the earlier section, it was observed that CLIP’s performance was suboptimal when considering both text and image encodings simultaneously. However, given CLIP’s robust image encoder, I proceeded to conduct a series of experiments to assess its classification performance independently.

I began by training a Multi-Layer Perceptron (MLP) on top of CLIP’s image encoder rather than directly engaging in full fine-tuning – this was a deliberate choice aimed at optimizing both computational efficiency and model performance. By leveraging the pre-trained features

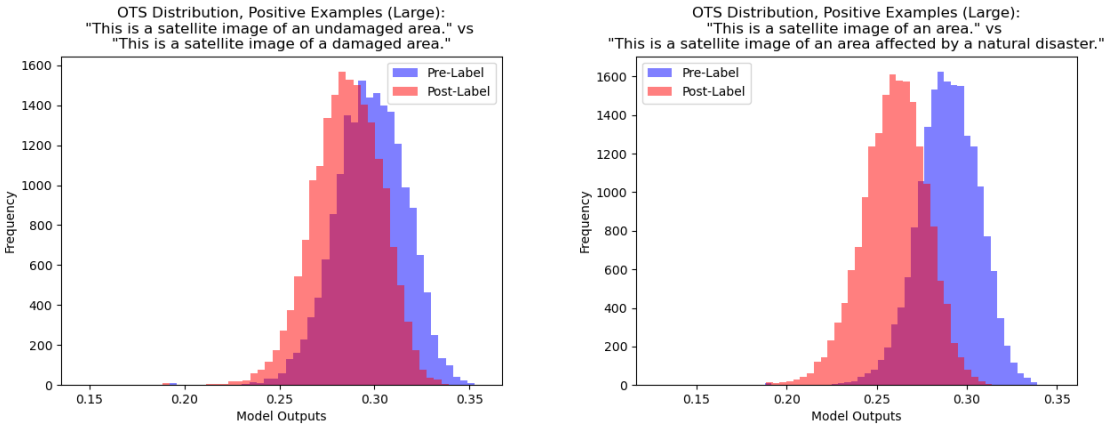

Figure 3.2: Distributions of CLIP class scores (pre-normalization) for two different prompts on post-disaster images.

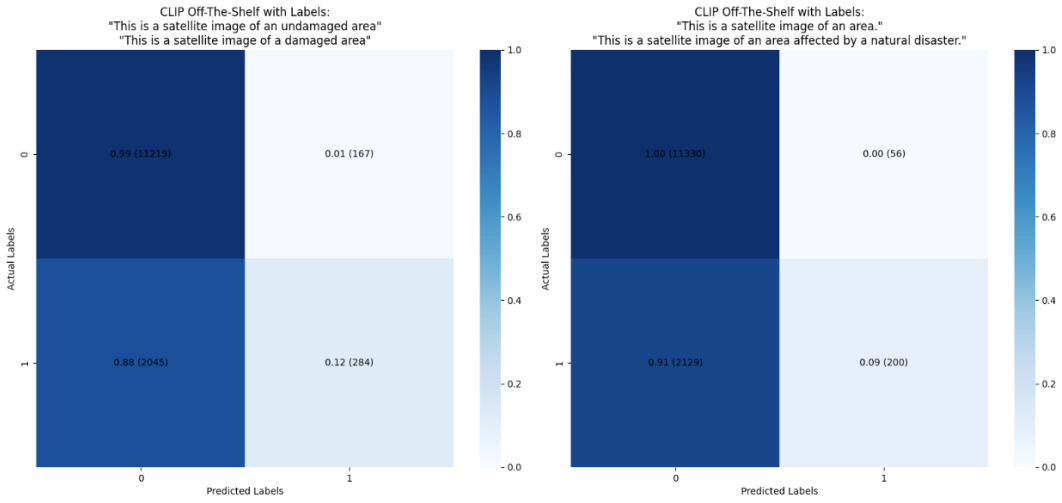

Figure 3.3: Confusion Matrices for Off-The-Shelf performance.

extracted by the image encoder, I aimed to harness the wealth of knowledge encoded within its parameters, which have been learned from vast amounts of diverse image data.

This approach performed poorly on differentiating between the post-disaster images for both the positive and negative classes on both in-distribution and out-of-distribution data. The results depicted in Figure 3.4 reveal the inadequacy of this approach in distinguishing between positive and negative images post-disaster. In both in-distribution and out-of-distribution scenarios, the resulting confusion matrices illustrate a significant misclassification between positive (post-disaster) and negative (pre-disaster) classes.

The findings depicted in Figure 3.4 suggest that this model may lack sufficient information or discriminative features to effectively distinguish between positive and negative images. This is evidenced by the notable misclassification rates observed in both in-distribution and out-of-distribution scenarios, where a substantial portion of post-disaster images is incorrectly labeled as positive (incurring damage). These results hint at the need for additional features or refined training strategies to enhance the model's capacity to discern between these images

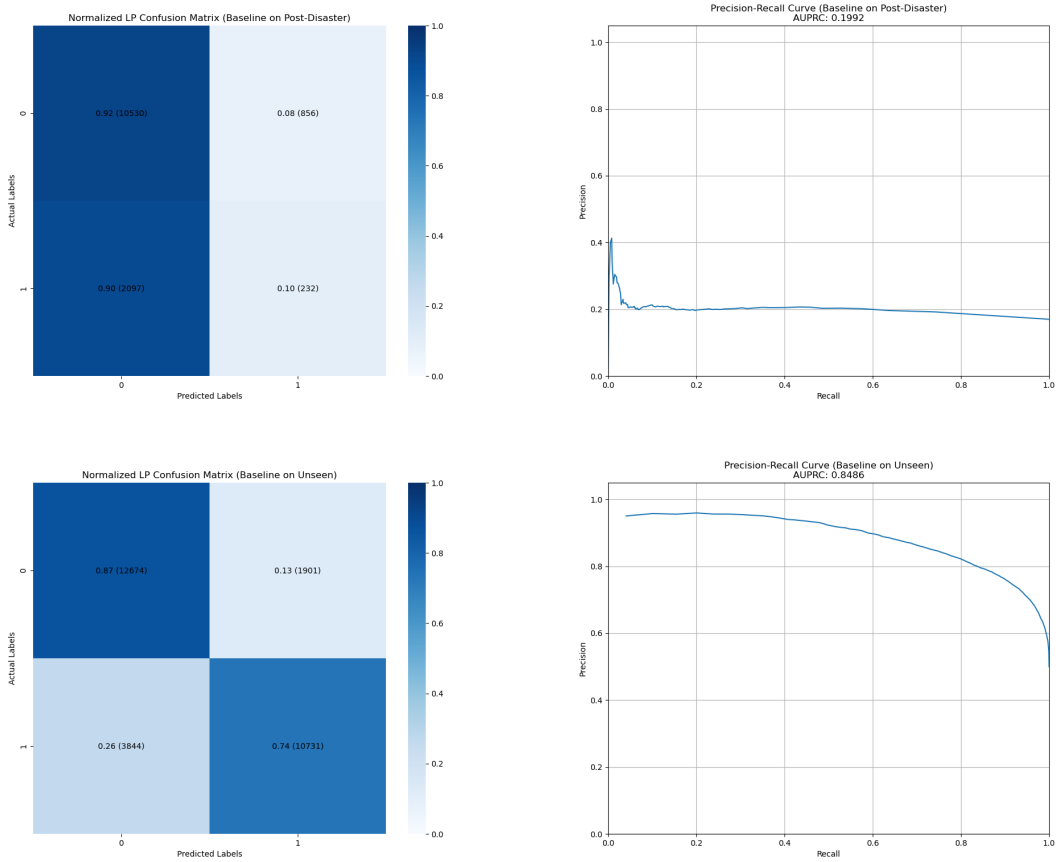

Figure 3.4: Distributions of CLIP class scores for two different prompts on post-disaster images.

more effectively.

3.4 Combining Image Encodings

To progress beyond linear probing, I explored methods leveraging paired image encodings - Figure 3.5 illustrates a straightforward pipeline for this approach. In this setting, the original CLIP image encoder is still being used, to encode pre- and post-disaster images. The two resulting encodings are then concatentaed together (to make one large paired encoding), before being fed into an MLP for classification. In this set of experiments, only the MLP is being trained (see Section 3.7 for experiments on fine-tuning the CLIP encoder as well). However, during experimentation, it became evident that testing data required redistribution due to a significant disparity in the number of negative and positive paired examples (approximately 91,000 vs. 18,000). Among various methods, random oversampling emerged as the most effective strategy.

In comparison to the earlier linear probing experiments, some improvements were observed, particularly when evaluating models on unseen data, as can be seen in Figure 3.6. However, in the evaluation of the unseen data, a significant number of "false negatives" still occur.

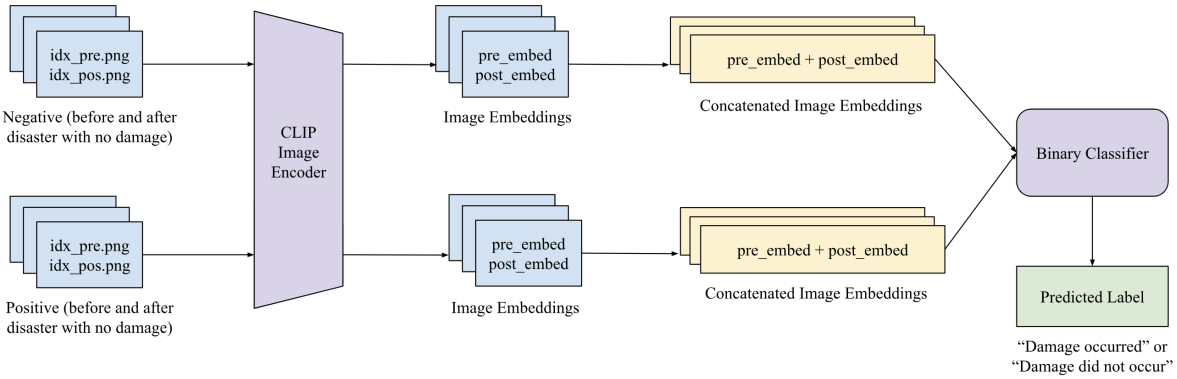

Figure 3.5: Pipeline demonstrating the use of CLIP with Paired Encodings

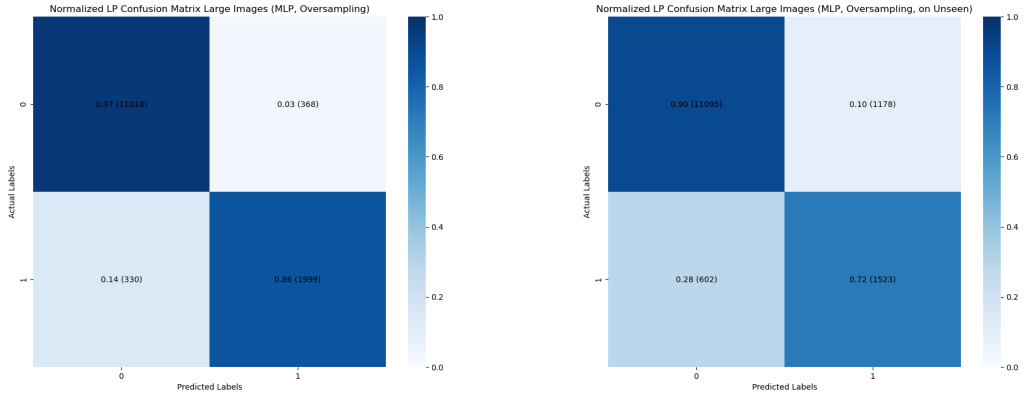

Figure 3.6: Combining Image Encodings on Tier 1 and Tier 3 Images

3.5 Combining Image Encodings and Text Encoding

In the previous section, we observed that incorporating additional information, particularly through including both pre- and post-disaster image encodings, proved to be beneficial for enhancing model performance. Building upon this insight, I sought to further augment my approach by integrating text encodings into the model's representations (in a different manner than the typical CLIP model). I anticipated that this combined approach would leverage the strengths of both modalities, allowing for a richer representation of the input.

For this approach, I tested a variety of disaster-related text prompts including examples like:

1. "This is a satellite image of an area damaged by a <disaster>"/"This is a satellite image of an area"
2. <disaster>
3. "Aerial image of landscape after a <disaster>"

Similar to the structure shown in Figure 3.5, these text prompts were tokenized and encoded, then concatenated to the paired encoding of the disaster image that they corresponded to. Figure 3.7 demonstrates an example of results from this approach. In comparison to the paired

image encodings approach, this approach performed significantly worse - especially in regards to the Tier 3 data, indicating that including this extra text information reduced the robustness of the resulting model. This trend could possibly be attributed to the difference in disaster types present in the Tier 3 data – corresponding text encodings would not have been seen while training on Tier 1 data.

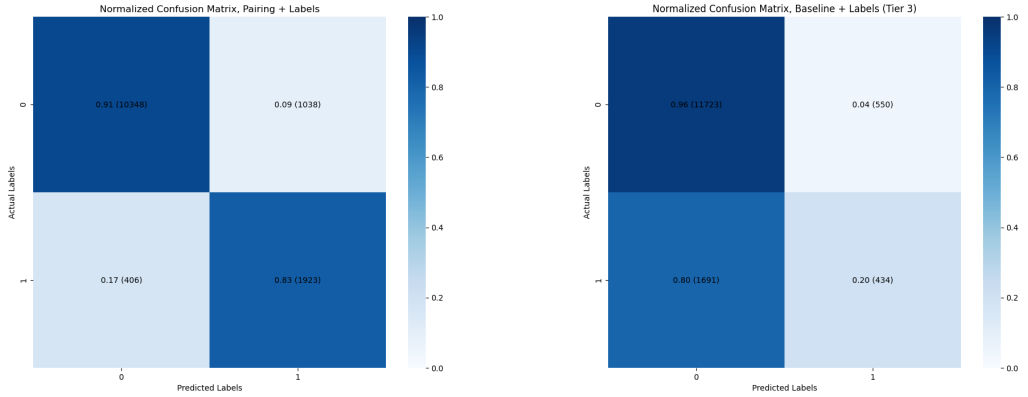

Figure 3.7: Combining Image and Text Encodings on Tier 1 and Tier 3 Images

3.6 Stacked Generalization for Groups of Disasters

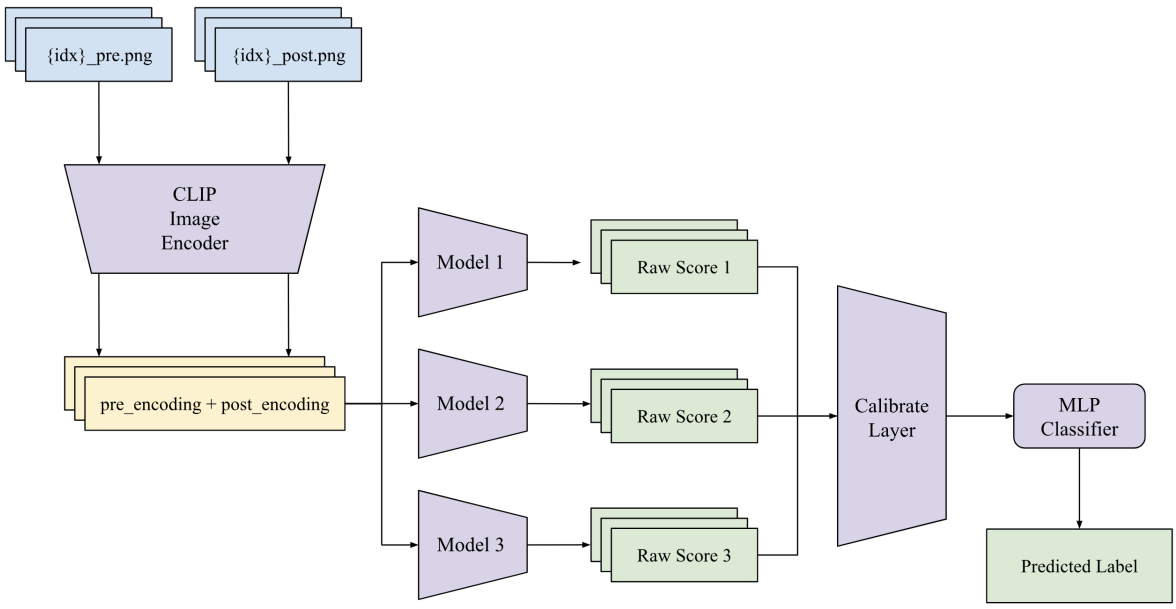

Figure 3.8: Pipeline Demonstrating Stacked Generalization Approach with Paired Encodings

To further improve performance, I transitioned to a stacked generalization approach. This methodology aims to leverage the collective intelligence of multiple specialized models to make more accurate predictions [7]. Traditional linear probing and paired image classification

methods encountered limitations in effectively capturing the nuanced features and patterns crucial for accurate classification - however, by incorporating this framework, I aimed to harness the complementary strengths of individual models specialized in different groupings of disasters.

For this stacked generalization approach, I trained three different models, on different groups of disasters, still utilizing the paired encoding approach covered in Section 3.4. As is demonstrated in Figure 3.6, an input image would go through all three models, resulting in three predictions, before a calibration layer converts these three predictions into one overall prediction.

3.6.1 Choosing Disaster Groupings

To motivate an ideal set of disaster groups, I began by training a model to determine disaster groups, then evaluating it on unseen data. Here, I aimed to identify common instances of misclassification to unveil recurring patterns that might indicate the best disasters to group together - as shown by Figure 3.9, patterns in misclassification indicated three possible groups.

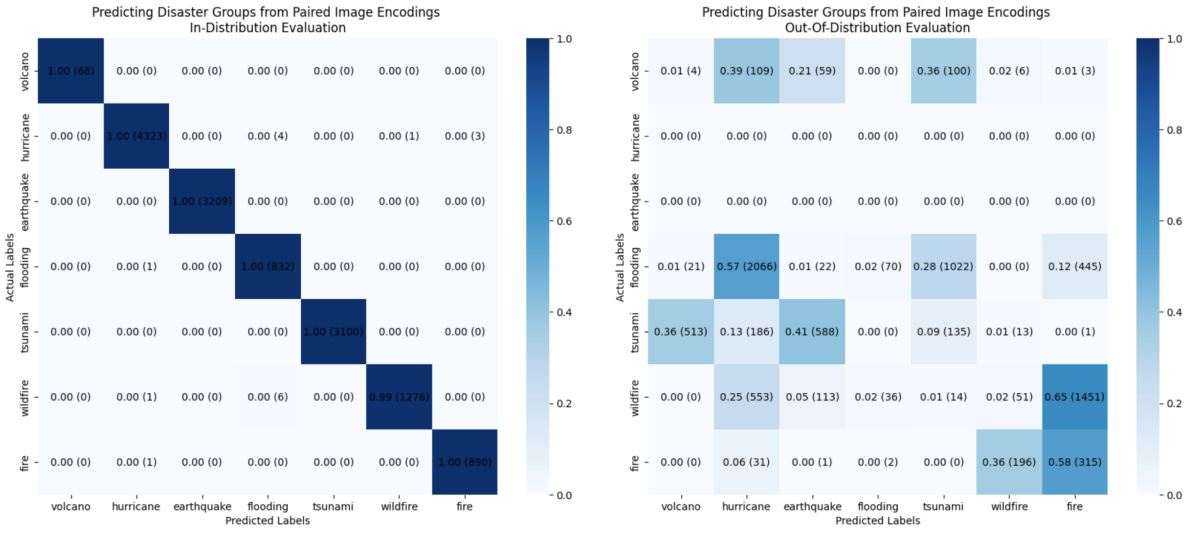

Figure 3.9: Classifying Disaster Type from Image

Attempting to use K-Means Clustering

My exploration into disaster grouping led to an experiment with a small k-means clustering as a potential solution for segmenting images based on feature similarity. Here, the motivation was that images could be grouped together in less obvious ways, outside of simple disaster grouping - then, models could be trained on the images in these clusters, and evaluation would involve assigning an image to a cluster (and its subsequent model). However, this attempt yielded less-than-ideal results - Figure 3.10 shows the clustering visualization projected in two dimensions using Principal Component Analysis (PCA), with 5 clusters. Each disaster type seems to have clear boundaries, but these boundaries for each disaster type heavily overlap.

The high dimensionality of the feature space, posed difficulties for k-means to accurately define meaningful distances between data points. Additionally, the complex and varied nature of

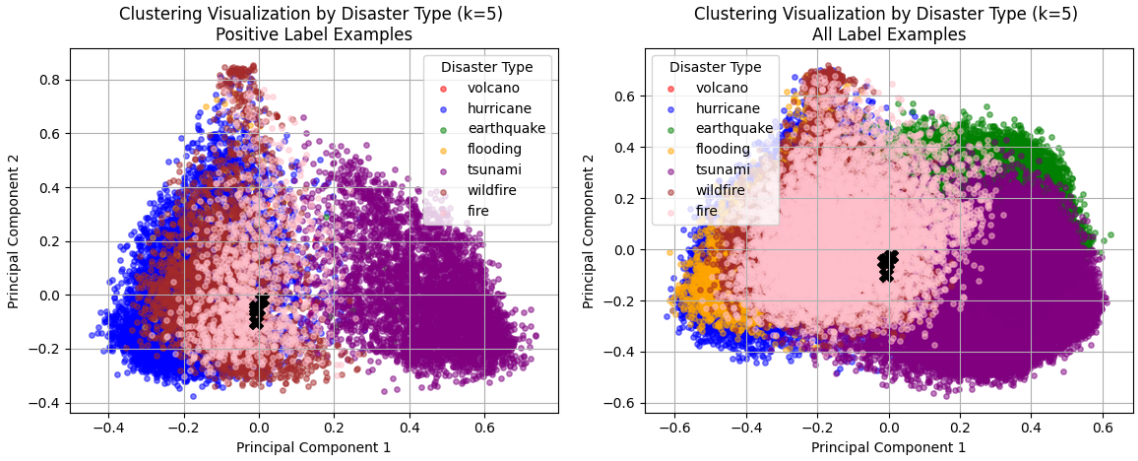

Figure 3.10: Attempt to Classify Disaster Type with K-Means

disasters resulted in clusters with irregular shapes and ambiguous boundaries, as well as an inability to create discernible clusters. This trend was seen for varying number of clusters.

Final Disaster Groupings Chosen

Finally, after testing out several disaster groups, the final grouping was chosen:

- Fire-Based Disasters: wildfire, fire, bushfire
- Water-Based Disasters: flooding, tsunami
- Extreme Weather Disasters: tornado, volcano, hurricane, earthquake

Note that "hurricanes" were determined to be an "Extreme Weather Disaster" rather than a "Water-Based Disaster" – this yielded better classification results, likely due to the patterns of damage that result from the disasters.

3.6.2 Calibrating Base Model Results

As mentioned earlier, in the stacked generalization framework, multiple base models are trained to make predictions based on the input data. Each base model is trained independently and may specialize in capturing different aspects or patterns within the data. When a data point arrives for prediction, it is fed through all of the base models, obtaining individual predictions from each, which are then fed as input to a stacking, or calibration, layer [7]. Having learned from the combined predictions in the training set, this layer uses its knowledge to produce the final prediction. In this section, I discuss a few approaches to creating this final layer.

Simple "OR" Combination

The most simplistic approach taken in the stacked generalization framework is to perform an "OR" combination of the predictions from the base models. In this approach, each base model produces a prediction in the form of probabilities for each class. These predictions are then thresholded such that if any of the base models predict a positive outcome, the final prediction is set to positive [7].

Previous approaches showed a high number of false negatives, especially when evaluation on unseen data. The "OR" approach tends to reduce the risk of false negatives by considering any positive prediction from the base models as evidence for a positive outcome. Additionally, each base model may have its strengths and weaknesses in capturing different aspects of the data or in handling noise. By considering the prediction of any base model that indicates a positive outcome, the combined model can be more robust to individual model variability.

Figure 3.11 shows the results of this approach, which are comparable to the paired image encoding results. While the number of false positive results increased, the number of false negative results decreased - a step in the right direction

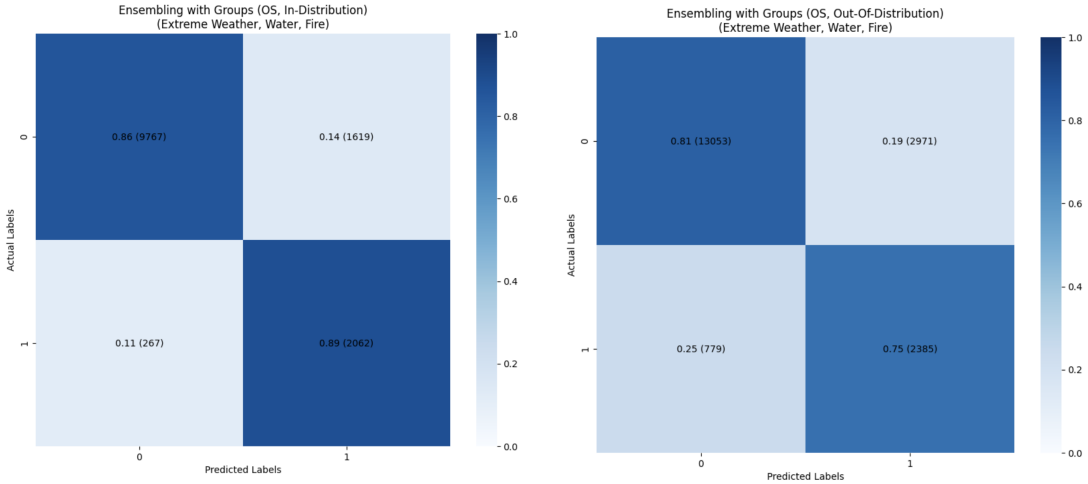

Figure 3.11: Stacked Generalization with "OR" Layer on Tier 1 and Tier 3 Images

To improve these results further, I attempted to calibrate them further by modifying the threshold values for each model (for reference, the earlier experiment used the common threshold of 0.5). Here, the threshold value refers to the 'cutoff' for assigning a class prediction to the model's raw probability output (if the output for some input is above the threshold, then that input is assigned to the positive class).

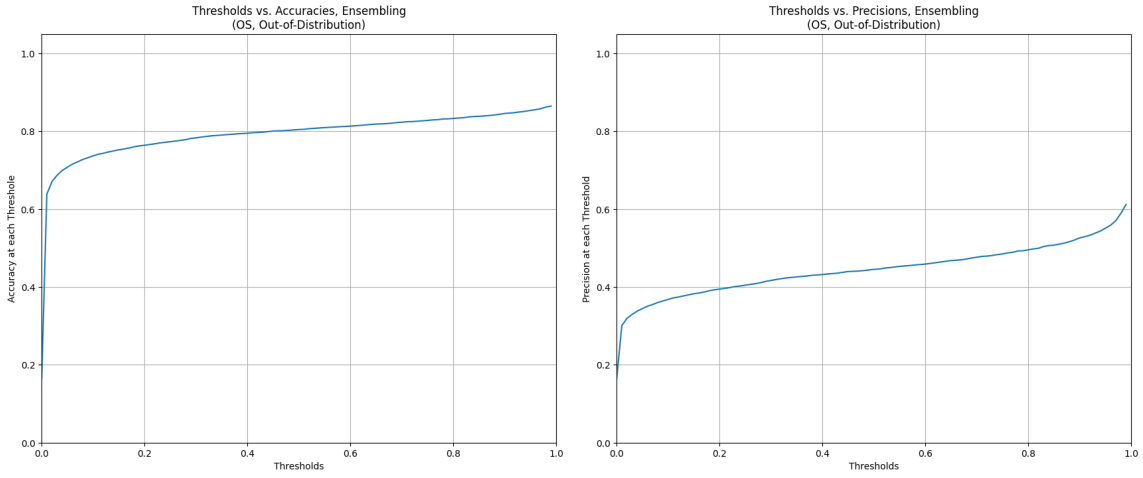

Figure 3.12: Threshold vs. Accuracy and Threshold vs. Precision Curve

As demonstrated in Figure 3.12, both the accuracy and precision seems to increase as the threshold increases. However, closer analysis demonstrated that this trend was slightly misleading – a higher threshold corresponded to a higher number of negative class assignments, which is the majority class in this problem. Ultimately, it was determined that the threshold of 0.5 was most suitable for decreasing the number of false positives, a more important goal.

Averaging Base Model Predictions

Another approach taken was to simply take the raw predictions from all the base models and calculate their average before thresholding. This is a simple approach, treating all models equally, regardless of their individual accuracy or bias [7]. Regardless, exploring this approach was motivated by its potential simplicity and intuitiveness. If successful, it would offer a straightforward solution that demands minimal additional training compared to constructing a separate stacking model. Directly combining all the base models' predictions simplifies the ensemble's decision-making process, providing clearer insights into how the final output is determined.

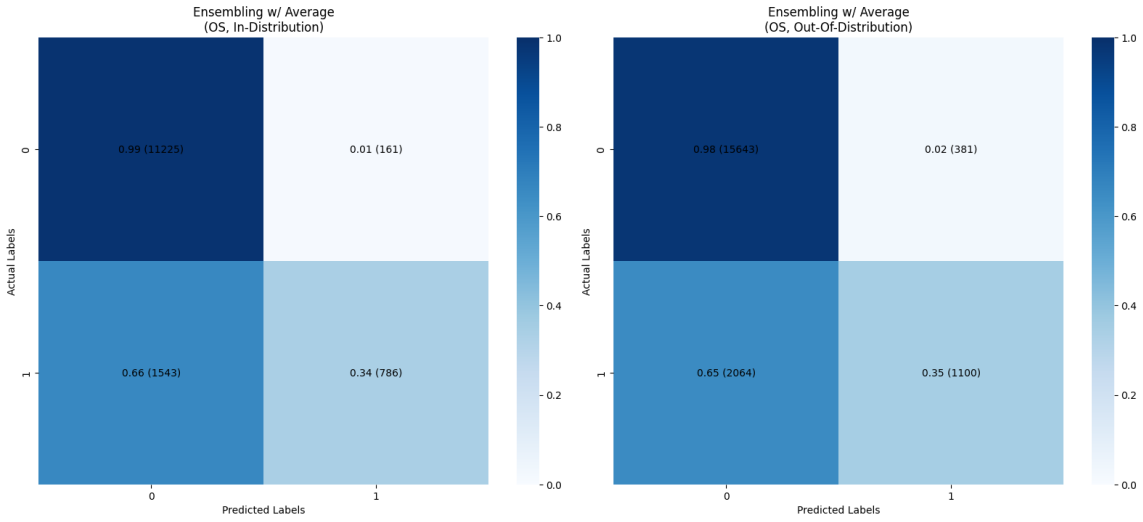

Figure 3.13: Stacked Generalization with Avg. Layer on Tier 1 and Tier 3 Images

As can be seen in Figure 3.13, this calibration approach performs poorly, resulting in a high number of false negatives. Trying a variety of thresholds led to the relatively same performance, indicating that this approach was too simplistic for this problem.

Logistic Regression Combination

Finally, I tried training a logistic regression head, that took in the raw outputs of the three models and returned the resulting class (after thresholding). As can be seen in Figure 3.14, these results are similar to the "OR" calibration method, but do not lead to significantly better results than the paired image encoding method.

3.6.3 Findings and Limitations

TODO: include a table about the results of the expert models on individual disasters
TODO: include rows about disaster-based results of the paired encoding model, Table 3.6.3

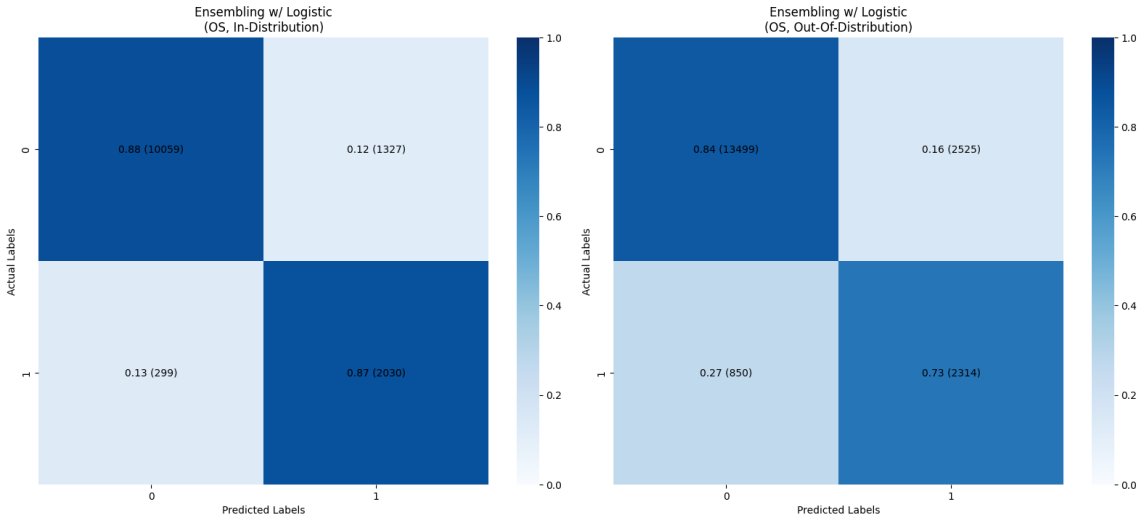

Figure 3.14: Threshold vs. Accuracy and Threshold vs. Precision Curve

Further evaluation showed that the base models in this approach themselves did not outperform the larger overall model (trained on all disasters). Table 3.6.3 demonstrates the accuracy of these models on a disaster-specific basis – close inspection shows that the base models do not significantly and regularly outperform the larger model, perhaps due to a lack of training data. Ultimately, this trend suggests that other approaches to the stacked generalization method are unlikely to result in an improvement in performance.

3.7 Full-Fine Tuning

Given the findings and limitations observed in the previous experiments, particularly regarding the limitations of the CLIP off-the-shelf model and the challenges faced in combining image and text encodings effectively, it became evident that further refinement was necessary to achieve the desired level of performance and robustness in the binary classification task. While earlier experiments focused on leveraging CLIP's pre-trained encoders as they were, they fell short in addressing the underlying issue of inadequate feature representation for distinguishing between positive and negative images.

It was clear that the next logical step was to explore the potential benefits of fine-tuning the CLIP vision encoder in conjunction with the classification head. By fine-tuning both components simultaneously, the model can learn to extract more discriminative features from the input images, tailored specifically to the classification task at hand. This approach offers several advantages over solely training the classification head on top of the pre-trained encoder [4].

3.7.1 Fine Tuning Image Encoder on Single Images

This process began with fine-tuning the parameters of the vision encoder, as well as training a classification head, on post-disaster images, allowing it to learn more task-specific features that are relevant for distinguishing between positive (damaged) and negative (undamaged) images, similar to the experiments in Section 3.3.

TBD: still waiting on evaluation job to finish, to get metrics

3.7.2 Fine Tuning Single Image Encoder on Paired Images

In this section, I look into fine-tuning the single-image encoder on paired images to enhance the performance of the resulting model. Leveraging the insights gained from previous experiments discussed in Section 3.4, where paired encodings were employed, I conducted a similar experiment while fine-tuning the CLIP visual encoder. Findings indicate that employing a lower learning rate yielded superior performance compared to higher learning rates, as suggested by other works [8].

TBD: table about accuracy, precision and recall for varying learning rates, Table 3.7.2

3.7.3 Fine Tuning Siamese Image Encoder Network for Paired Images

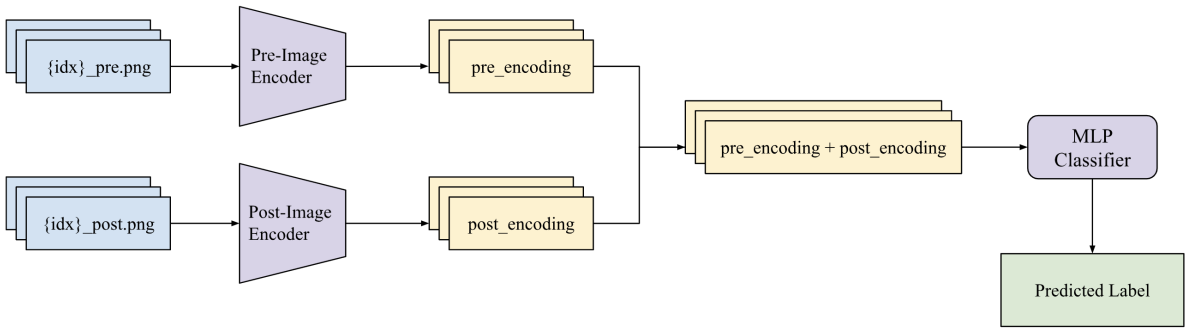

Figure 3.15: Fine-Tuning using Paired Images with Siamese Architecture

Fine-tuning a Siamese image encoder network for paired images involves training a neural network architecture specifically designed to compare and extract features from pairs of images. The Siamese architecture typically consists of two identical subnetworks, each processing one image from the pair independently. These subnetworks are trained simultaneously on pairs of input images. Ultimately, by producing separate encodings for each image in the pair, the network can discern similarities and differences between them [9]. As demonstrated in Figure 3.15, two separate image encoders (based on CLIP's image encoder) are fine-tuned, one on pre-disaster images and one on post-disaster images. The encodings from the two separate encoders are concatenated and fed into an MLP (which is also being trained), to come up with a classification.

Just as was seen in the previous section, decreasing the learning rates corresponded to an increase in the performance and robustness of the resulting model, as is demonstrated in Table 3.7.3.

TBD: table about accuracy, precision and recall for varying learning rates, Table 3.7.3

Training ViT Encoder from Scratch

Exploring alternative approaches, I sought to determine if training the model from scratch on the Vision Transformer (ViT) encoder, instead of fine-tuning the CLIP ViT encoder, would yield any improvements in performance. By initiating training from scratch, the model undergoes the learning process without relying on pre-existing knowledge from the CLIP model. This allows for the exploration of potentially different feature representations and learning dynamics, which may better align with the specific requirements of the binary classification task.

However, when training a model from scratch, especially a deep neural network like ViT, there is a significant reliance on the capacity of the model to learn meaningful representations directly from the raw input data. Unlike fine-tuning, where the model starts with pre-trained weights that capture general visual knowledge, training from scratch requires the model to learn such representations solely from the training data. This can result in poor performance, especially when dealing with limitations like a small dataset [10]. As can be seen in Figure 3.16, training this encoder from scratch led to extremely poor performance on both in-distribution and out-of-distribution data, especially in comparison to the performance of the same structure when fine-tuned on CLIP's pre-trained vision encoder.

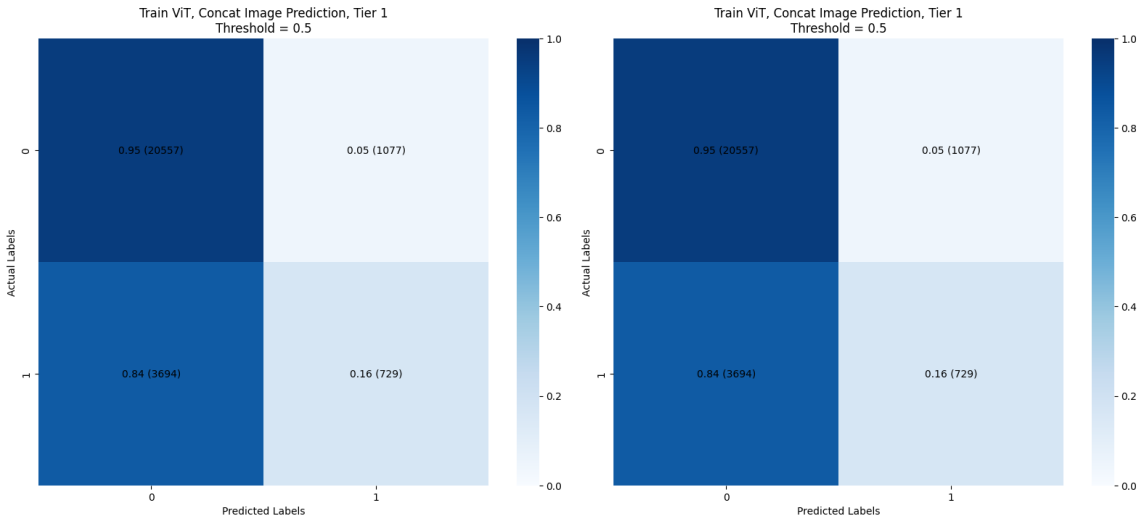

Figure 3.16: Results with Siamese Architecture with Base Vit Encoder on Tier 1 and Tier 3 Images

3.7.4 Fine Tuning Image and Text Encoder on Single Images

Finally, I attempted to see if any improvements could be gained by fine-tuning the text encoder along with the image encoder, by utilizing the same variety of text prompts that were used in the original analysis. However, as is demonstrated in Figure 3.17, even after fine-tuning the both the text and image encoder, the resulting model was unable to separate the two classes, similarly to the trends seen in Section 3.2.

TBD: more evaluation results on different prompts with disaster-specific prompts

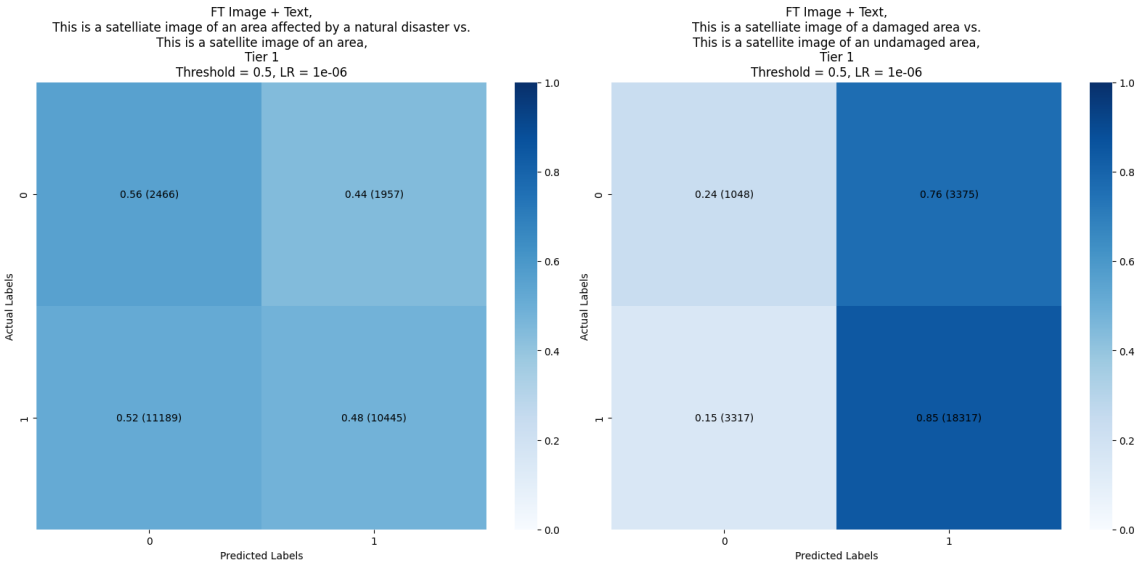

Figure 3.17: Results with Siamese Architecture with Base Vit Encoder on Tier 1 and Tier 3 Images

Chapter 4

Future Work

Foremost, it would be beneficial to explore the potential of fine-tuning the CLIP encoder with alternative architectures that were experimented with in this study. For instance, the stacked generalization, which demonstrated promising results when fine-tuning the CLIP vision encoder for paired images, could benefit from further refinement, possibly allowing for better performance from each of the sub-models, particularly when tailored to the specific nuances of disaster image classification. Additionally, investigating the feasibility of fine-tuning the CLIP encoder with other neural network architectures, such as attention-based models or hybrid architectures combining convolutional and transformer networks, could provide valuable insights into optimizing feature extraction for this task. By systematically exploring and fine-tuning the CLIP encoder with a range of architectures, future research can deepen our understanding of effective strategies for leveraging CLIP in disaster image classification applications.

Additionally, the exploration of semi-supervised learning methods, such as FixMatch, presents a promising avenue for enhancing the robustness and generalization capabilities of the binary classifier. FixMatch, a state-of-the-art semi-supervised learning algorithm, combines labeled and unlabeled data during training to improve model performance. By leveraging a large pool of unlabeled data, FixMatch enables the model to learn more robust representations of the input space, leading to enhanced classification accuracy and generalization to unseen data [11].

Integrating FixMatch into the training pipeline could offer several advantages for disaster image classification tasks. Firstly, it could alleviate the reliance on large labeled datasets, which are often scarce and expensive to obtain, by leveraging the abundant unlabeled data typically available. This would enable the model to learn from a broader range of examples, capturing diverse patterns and variations present in disaster images. Additionally, FixMatch's consistency regularization mechanism encourages the model to produce consistent predictions for augmented versions of the same unlabeled image, thereby promoting more robust and stable feature representations, helping to counter some of the issues caused by distribution shift.

Chapter 5

Conclusion

Throughout this thesis, I have explored various approaches to harnessing OpenAI's CLIP model as the foundation for a robust binary classifier, aiming to distinguish between damaged and undamaged areas in aerial images following natural disasters. While initial investigations, including off-the-shelf CLIP models, displayed promise, they often fell short in generalizing effectively to unseen datasets or accurately classifying images within our specific context. Despite efforts to enhance performance through fine-tuning the CLIP visual encoder and integrating text encoders into the classification pipeline, persistent challenges remained in achieving robust performance across different disaster types and datasets. The incorporation of text encodings, although attempted concurrently with image fine-tuning, introduced complexities that hindered the model's ability to discriminate between positive and negative classes effectively. These findings underscore the need for further exploration and refinement of methodologies to address the unique challenges posed by disaster image classification, with potential avenues including nuanced model architectures and more tailored training strategies. Ulimately, providing additional information for, as well as fine-tuning, the CLIP visual encoder yielded some improvements, particularly when employing lower learning rates.

Bibliography

[1] *Hurricane Costs*. National Oceanic and Atmospheric Administration (NOAA), 2024. [Online]. Available: https://coast.noaa.gov/states/fast-facts/hurricane-costs.html (visited on 04/25/2024).

[2] S. Kulinski and D. I. Inouye, *Towards explaining distribution shifts*, 2023. arXiv: 2210.10275 [cs.LG].

[3] Y. Lei, G. Sheng, F. Li, Q. Gao, C. Deng, and Q. Li, *High-discriminative attribute feature learning for generalized zero-shot learning*, 2024. arXiv: 2404.04953 [cs.CV].

[4] A. Radford, J. W. Kim, C. Hallacy, *et al.*, *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV].

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, *An image is worth 16x16 words: Transformers for image recognition at scale*, 2021. arXiv: 2010.11929 [cs.CV].

[6] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, and M. Gaston, *Xbd: A dataset for assessing building damage from satellite imagery*, 2019. arXiv: 1911.09296 [cs.CV].

[7] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992, ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(05)80023-1.

[8] X. Dong, J. Bao, T. Zhang, *et al.*, *Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vit-b and vit-l on imagenet*, 2022. arXiv: 2212.06138 [cs.CV].

[9] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network.," *IJPRAI*, vol. 7, no. 4, pp. 669–688, 1993.

[10] H. Zhu, B. Chen, and C. Yang, *Understanding why vit trains badly on small datasets: An intuitive perspective*, 2023. arXiv: 2302.03751 [cs.CV].

[11] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, *Fixmatch: Simplifying semi-supervised learning with consistency and confidence*, 2020. arXiv: 2001.07685 [cs.LG].