Weather Image Classification for Extreme Weather Events

Ariel Siman Tov | Tal Klein | Ido Vila

Group 24

Motivation

- Climate Change & Rising Risks
- Visual Data is Abundant
- Real-Time Detection = Faster Response

Accurate weather classification supports:

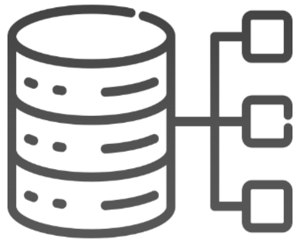
- Emergency services & infrastructure planning
- Transportation safety
- Insurance assessments
- Public warning systems

Our Project Goal

Build a reliable and interpretable model that classifies extreme weather from images – enhancing early detection, situational awareness, and decision-making.

What Makes Our Project Unique

We go beyond standard image classifiers like R-CNN by enhancing accuracy through multimodal learning :

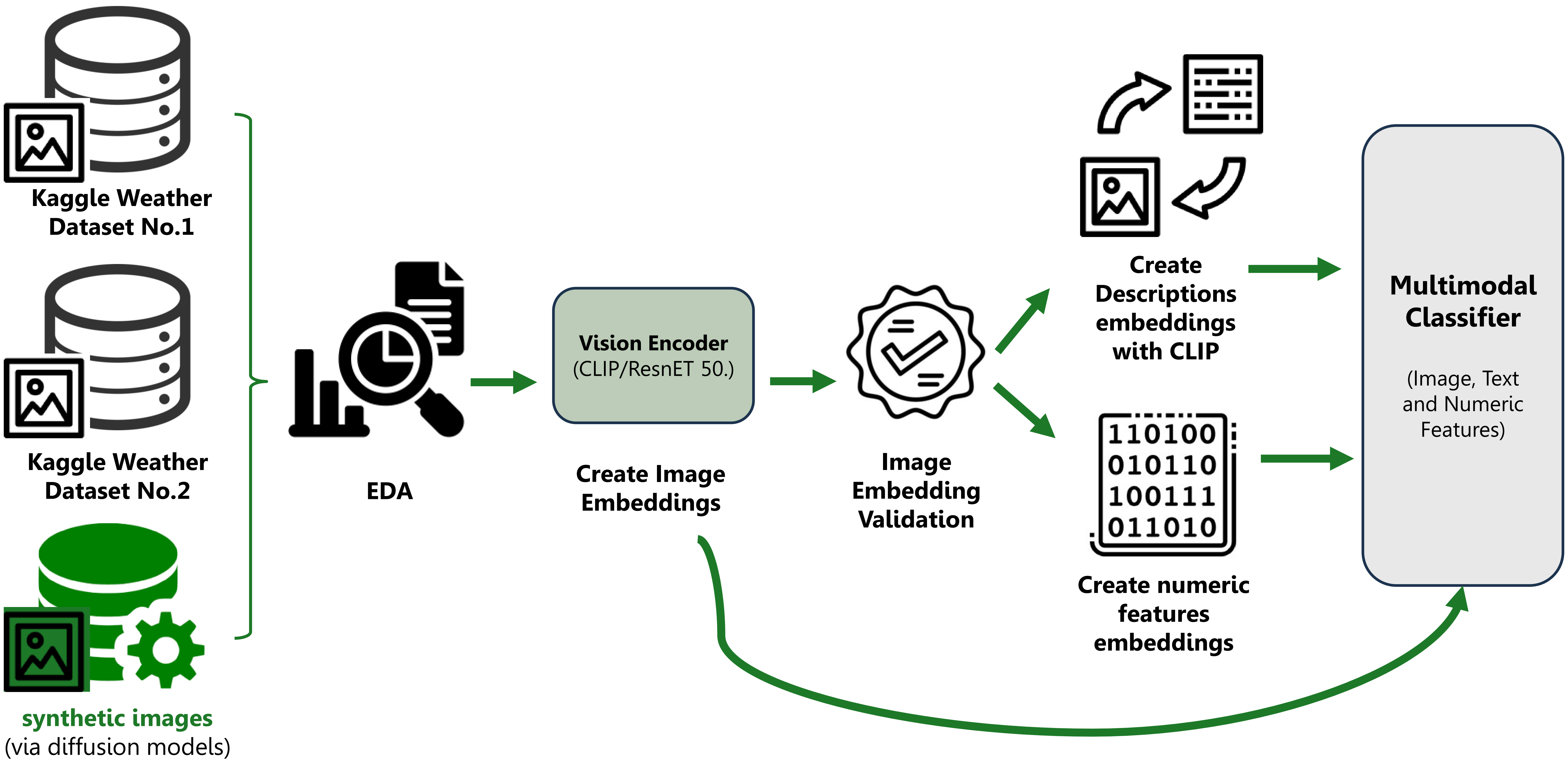


Expanded Dataset - Combined real and synthetic images (via diffusion model) for broader coverage of weather and disaster scenarios.

Embedding Validation – Verified synthetic image quality using t-SNE and cosine- similarity to ensure accurate class alignment.

Multimodal Learning - Integrated visual features, CLIP-generated textual descriptions, and numerical data for richer, more robust classification.

Our Workflow Pipeline



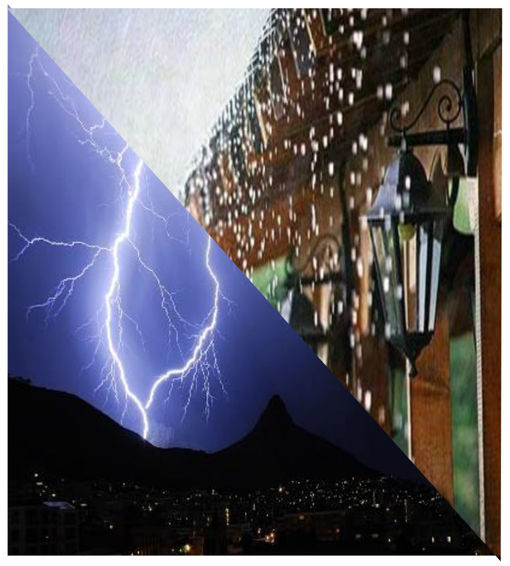
Generate New Wheater Classes Via diffusion model

We used two types of diffusion model configurations to synthesize new images:

Pipe 1 — A Prompt based diffusion model

Pipe 2 — An Image + Prompt based diffusion model

We wanted to create contrasting classics of natural phenomena, one that represents a natural disaster and one that does not:



(Skip to Colab-Notebook)

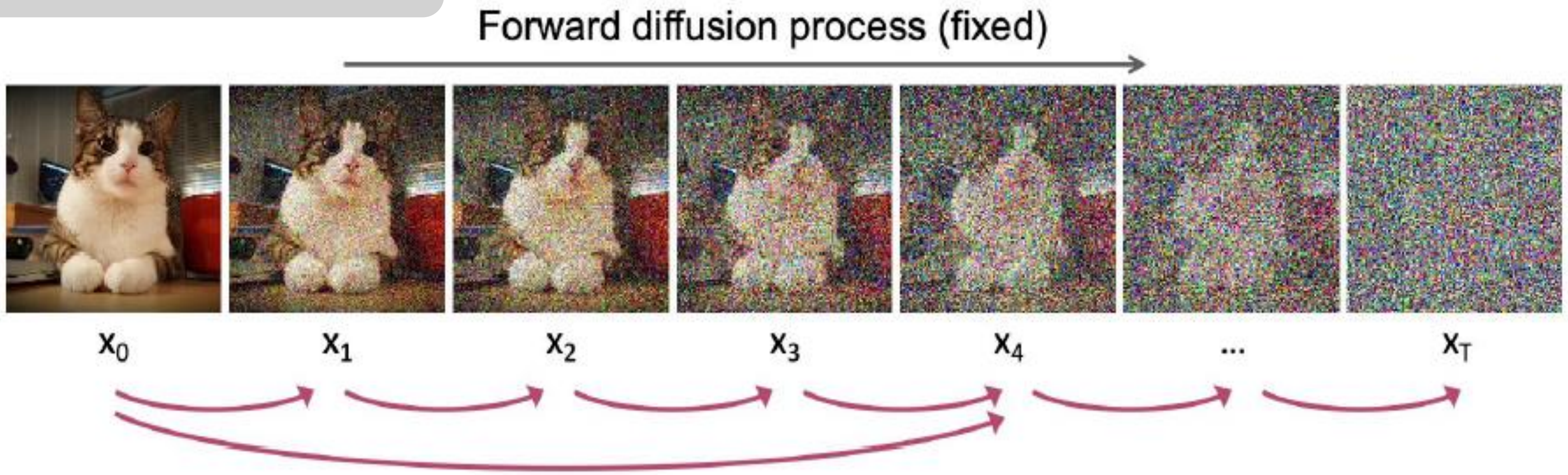
How Diffusion Models Generate Images

Diffusion models are generative models that learn to create data (like images) by reversing a noising process.

They consist of two main phases:

Forward Process
(Training Time)

Data



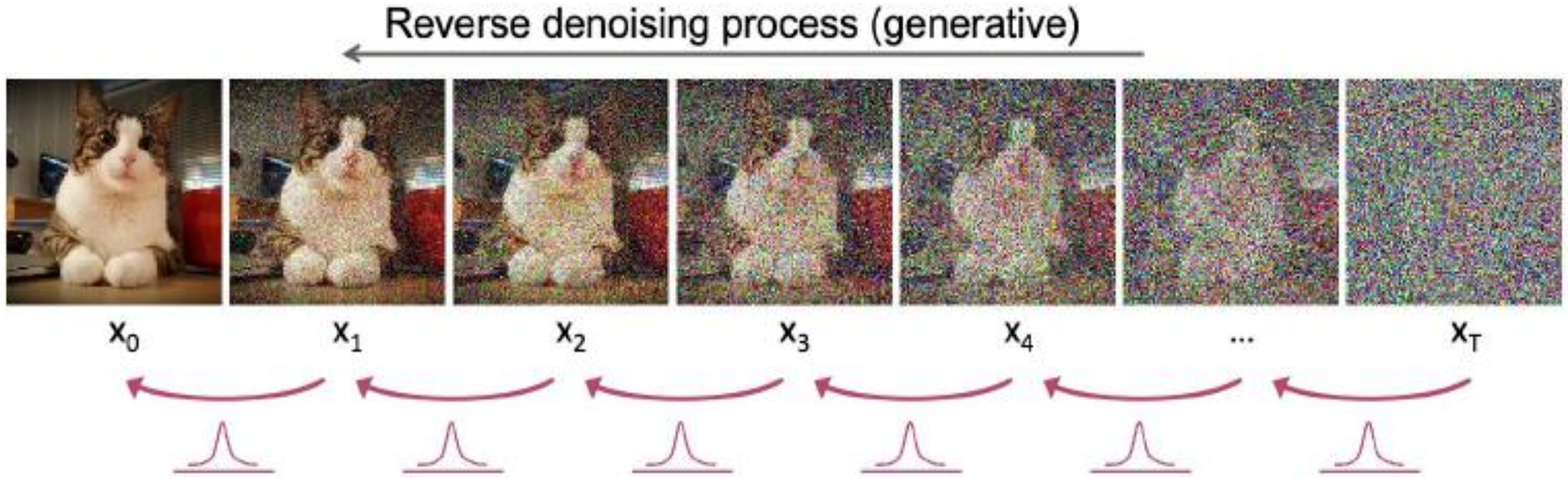
Noise

The Main Mission of the Forward Process :

the model is trained to **predict and remove the noise** added at each step

Reverse Process
(Generation Time)

Data



Noise

The Main Mission of the Reverse Process :

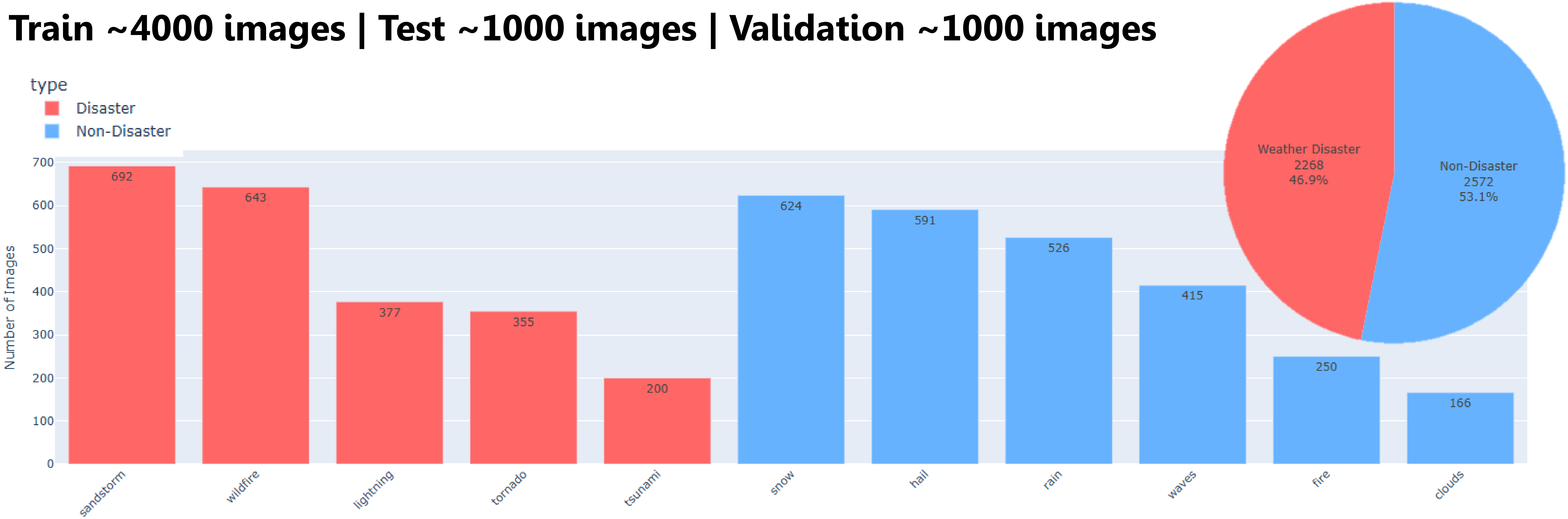
the model is trained to **predict and remove the noise** added at each step

Return

About The Data

Each Wheater category has between 200-700 images – total 4840 images

Train ~4000 images | Test ~1000 images | Validation ~1000 images



We aimed to achieve a balanced distribution of images across individual classes as well as between the broader categories of natural disasters and non-natural disasters

Exploratory Data Analysis (EDA)

(Skip to Colab-Notebook)

Embedding Validation by t-SNE

Intra-class Similarity — High

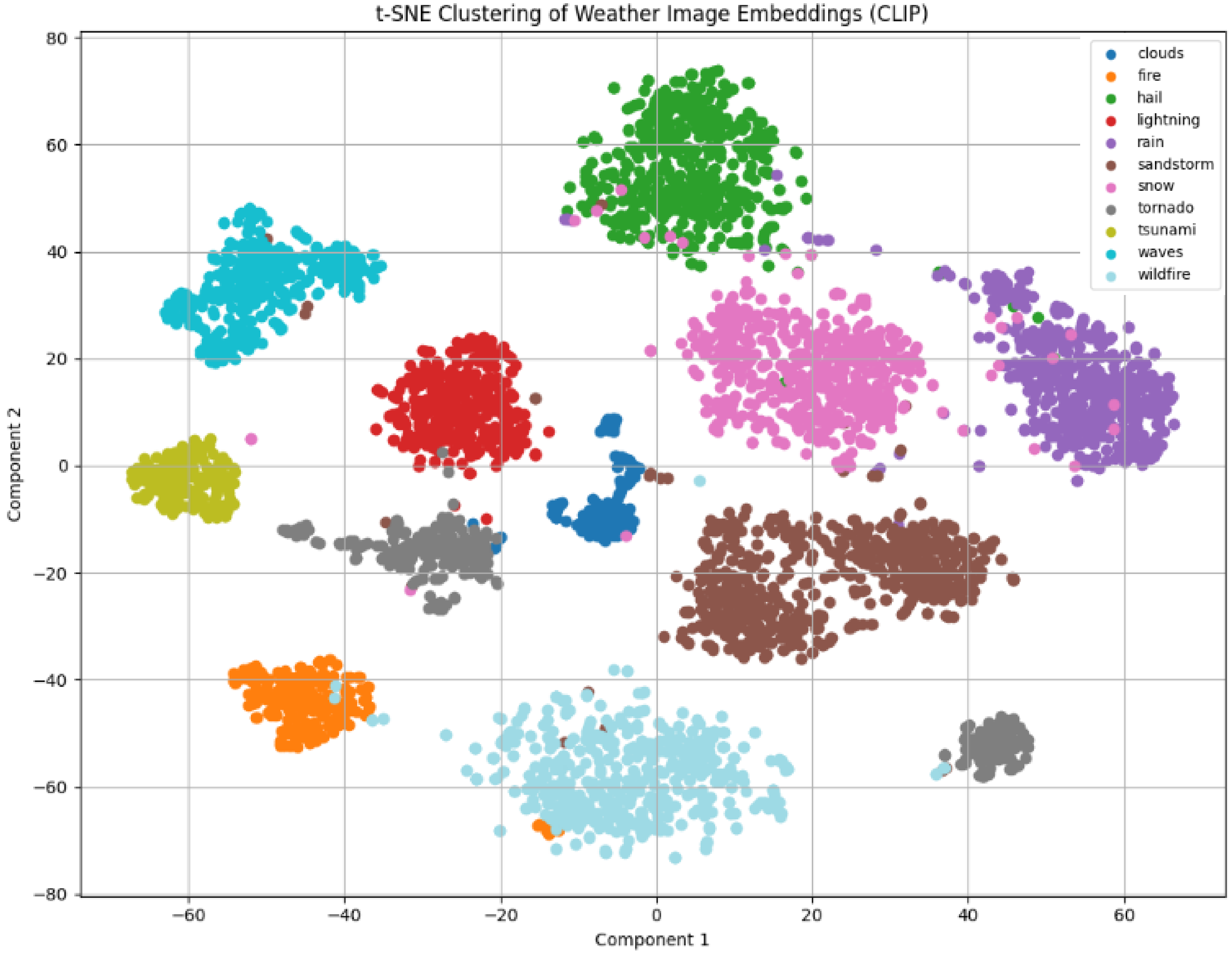
points forming cohesive clusters of weather phenomena

Intra-class Diversity — High

Classes exhibit healthy internal variation rather than appearing as single points

Inter-class Separability — Mid-High

Most classes form distinct well-separated clusters, although there are some concerning proximity between: Hail and lightning ,
Rain and sandstorm in some regions



Embedding Validation By Cosine-Similarity

Intra-class Similarity — High

Inter-class Relationships — Mid

<u>Highest similarity pairs</u> (potential confusion areas):
Tornado <-> lightning (0.88)
Clouds <-> tornado (0.86)
Clouds <-> waves (0.86)

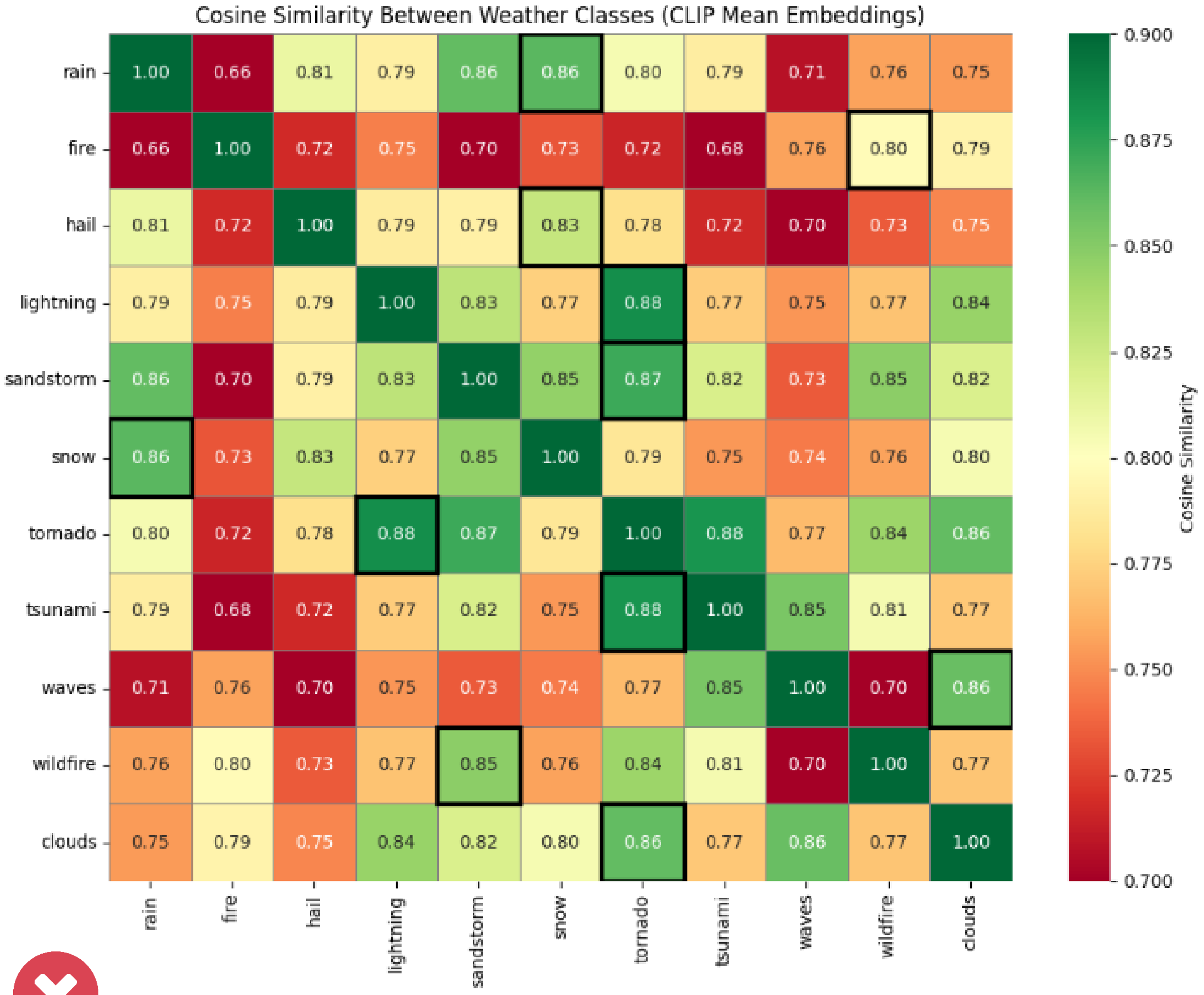
<u>Lowest similarity pairs:</u>
Fire <-> rain (0.66),
Fire <-> tsunami (0.68)
Wildfire <-> waves (0.70)

Class Distinctiveness Analysis

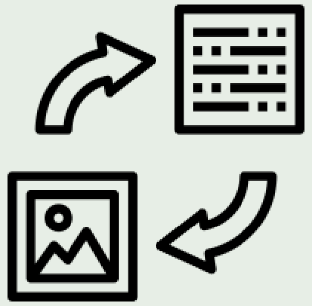
VS

Fire is the <u>most distinctive</u> class

Clouds show high similarity to several classes



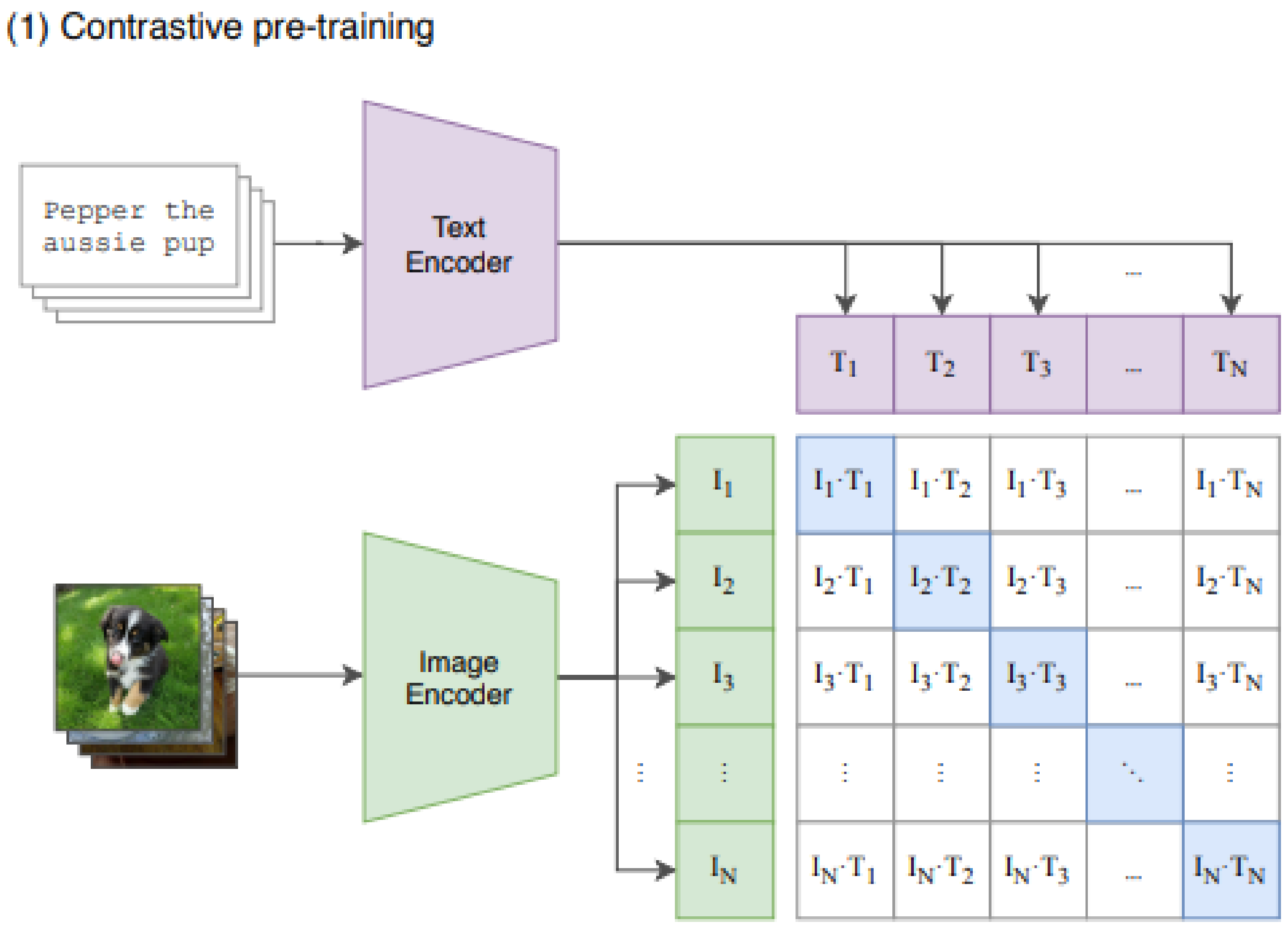
Create Descriptions Embeddings Via CLIP



What we have:

Image Embeddings encoded by CLIP

List of weather phenomenon descriptions



1) CLIP was trained on millions of image-text pairs

The **Image Encoder** (green) processes images into embeddings

The **Text Encoder** (purple) processes captions into embeddings

During training, **CLIP learns to make corresponding image-text pairs** have similar embeddings while making non-corresponding pairs dissimilar

Create Descriptions Embeddings Via CLIP

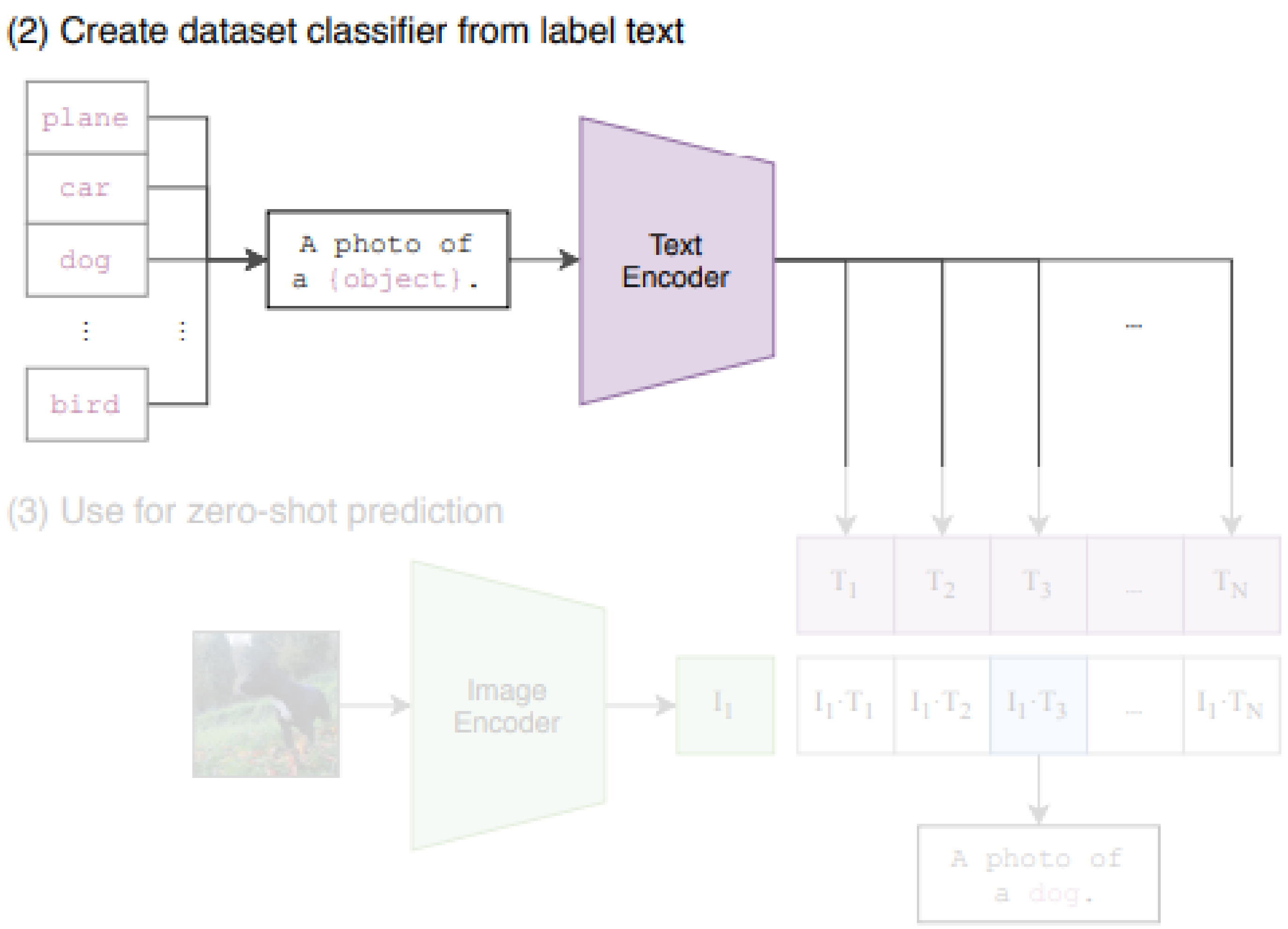
What we have:

Image Embeddings encoded by CLIP

List of weather phenomenon descriptions

2) Creating the Weather Description Classifier
providing **text descriptions** of weather phenomena: "sunny", "rainy", "cloudy", "snowy", etc.

The **Text Encoder** converts each weather description into text embeddings

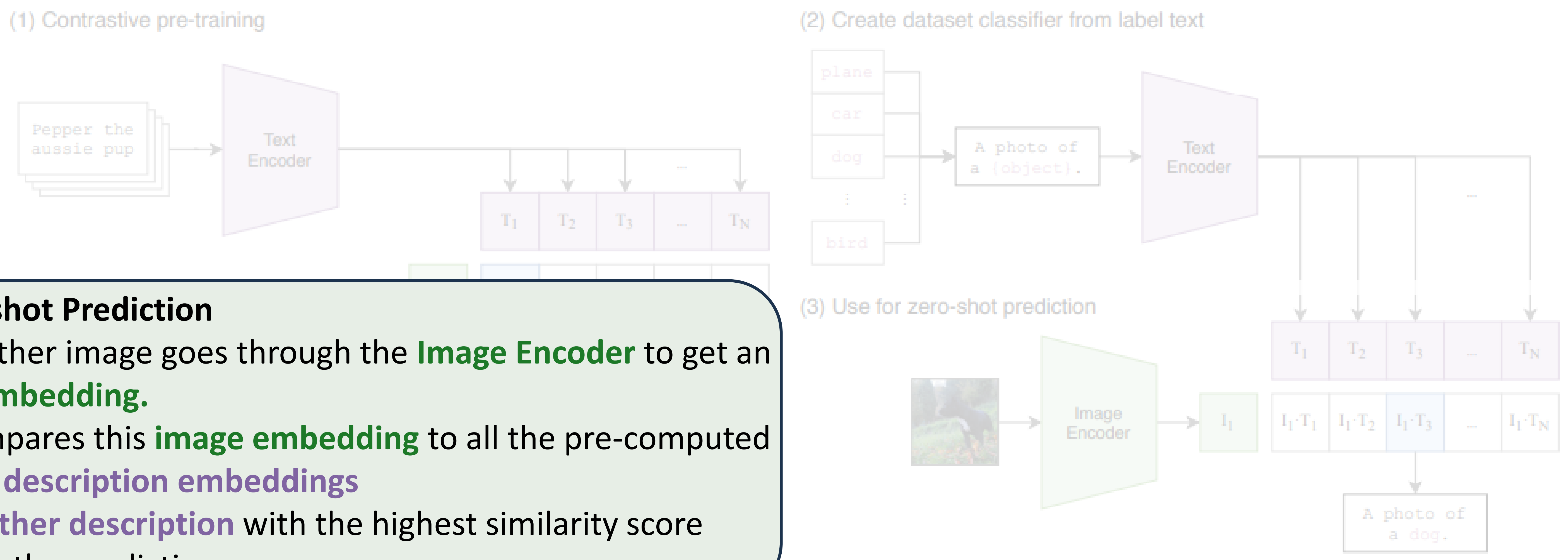


Create Descriptions Embeddings Via CLIP

What we have:

Image Embeddings encoded by CLIP

List of weather phenomenon descriptions



3) Zero-shot Prediction

The weather image goes through the **Image Encoder** to get an **image embedding.**

CLIP compares this **image embedding** to all the pre-computed weather **description embeddings**

The **weather description** with the highest similarity score becomes the prediction

Image Numeric Features

brightness / contrast

Calculated from pixel values, these features capture the overall lightness and tonal variation of the image

They are often indicative of specific weather phenomena (e.g., dark storms vs. bright snow)

avg_r, avg_g, avg_b

These values represent the average intensity of red, green, and blue channels, respectively

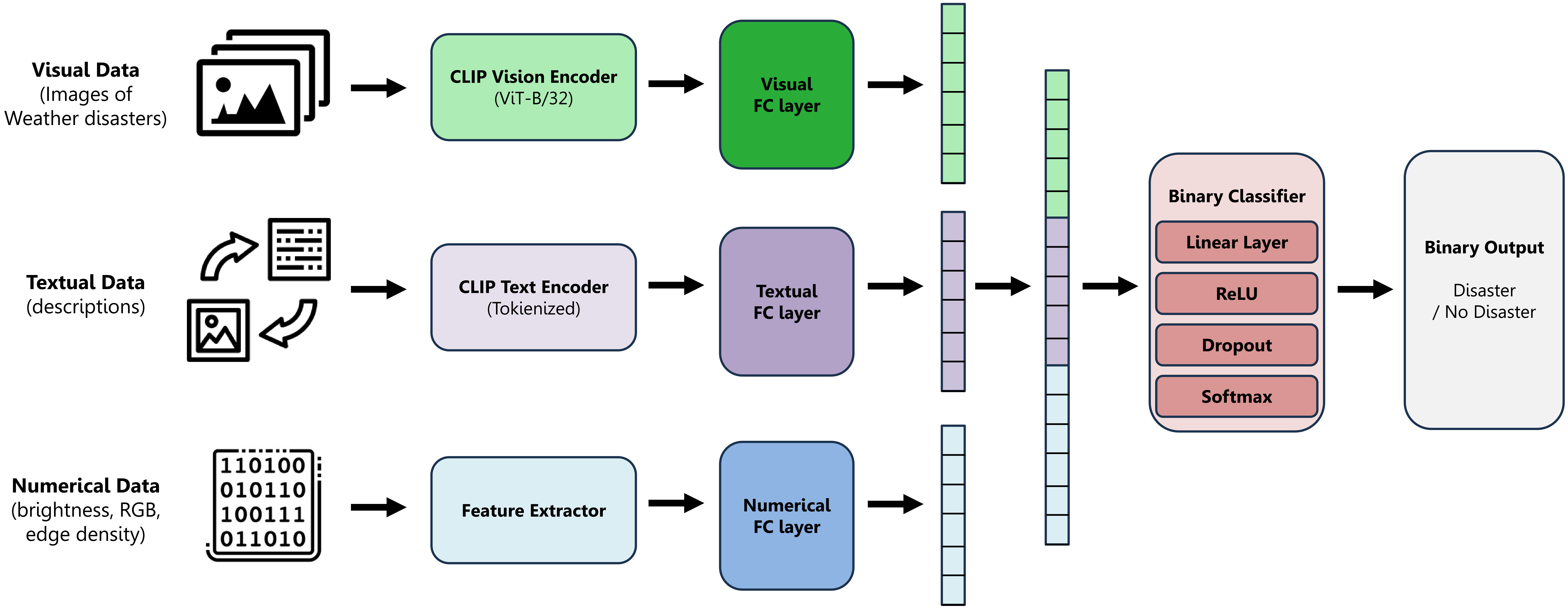
They provide a summary of the image's dominant color palette.

edge_density

This feature quantifies the amount of edge information in an image, computed using edge detection techniques

It reflects texture and structure, which can help distinguish between smooth phenomena (like clouds) **and chaotic ones** (like wildfires or tornadoes)

Our Multimodal Classifier Architecture



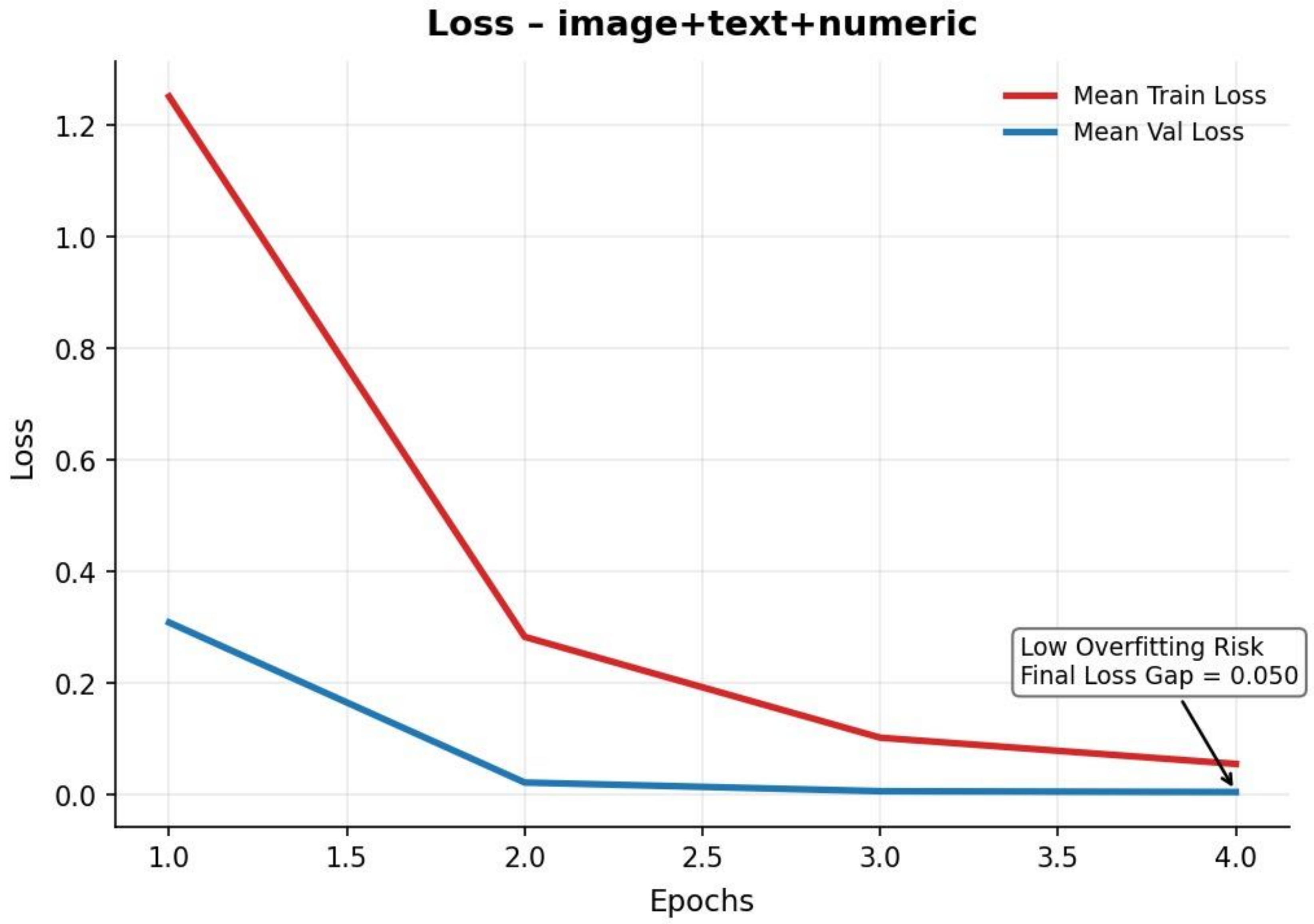
(Skip to Colab-Notebook)

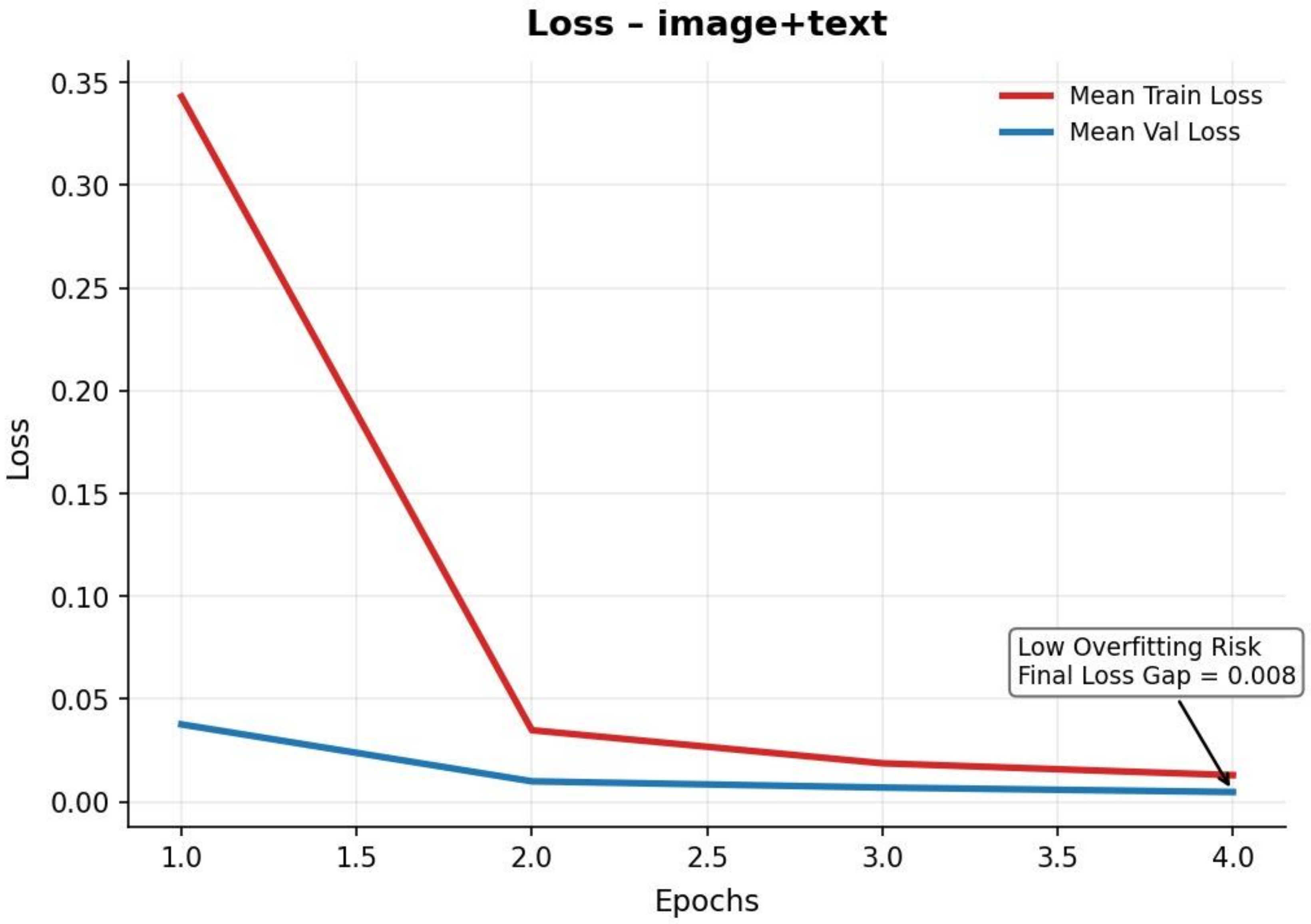
Evaluation

Performance Comparison Results throw modalities

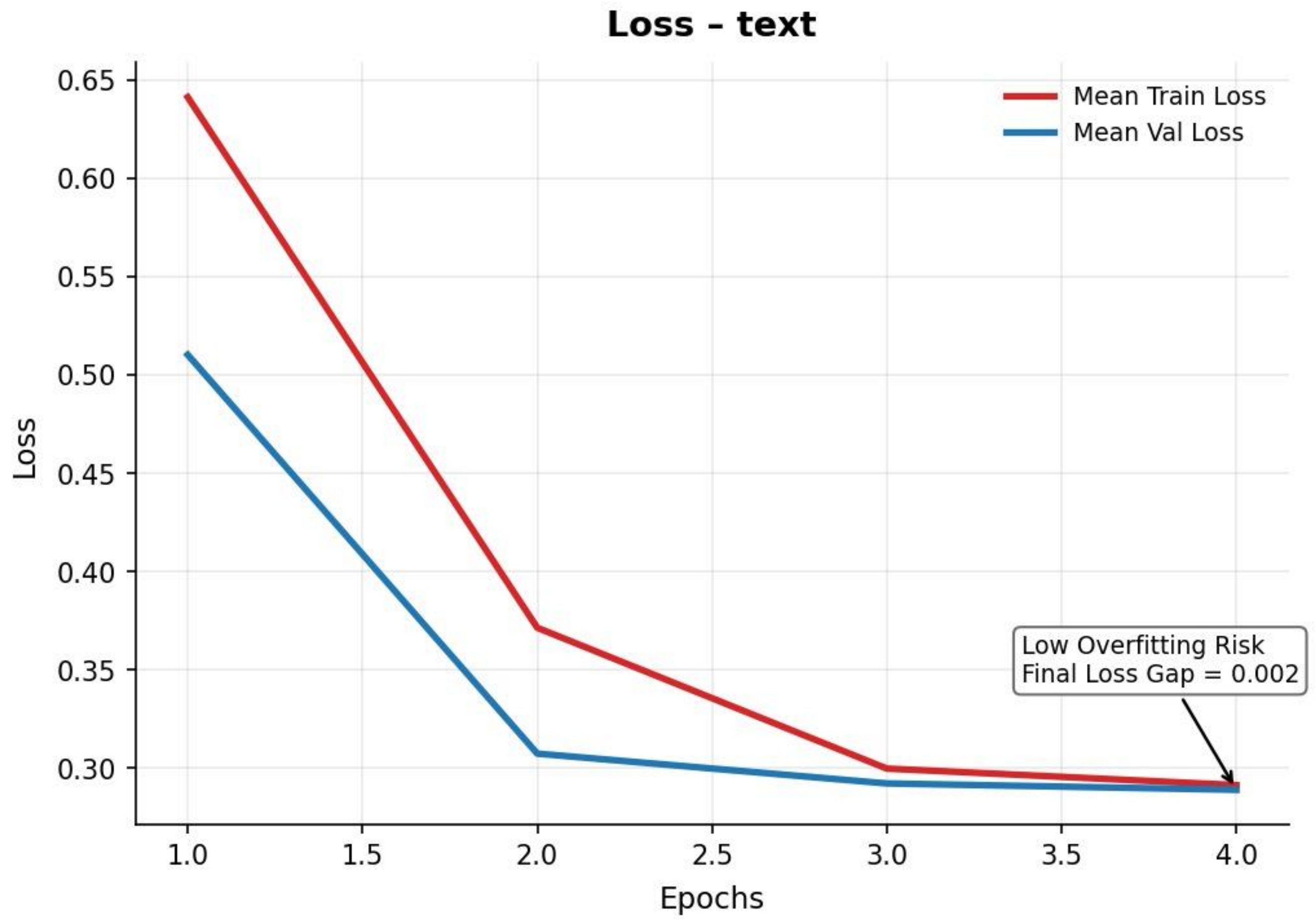
Modality Combination	Accuracy	Precision	Recall	f1
Image + text + numeric	0.997	1	0.993	0.996
Image	**1**	**1**	**1**	**1**
Text	0.898	0.855	0.925	0.888
Numeric	0.699	0.733	0.5	0.594
Image + text	**1**	**1**	**1**	**1**
Image + numeric	0.997	0.993	1	0.996
Text + numeric	0.909	0.871	0.931	0.9

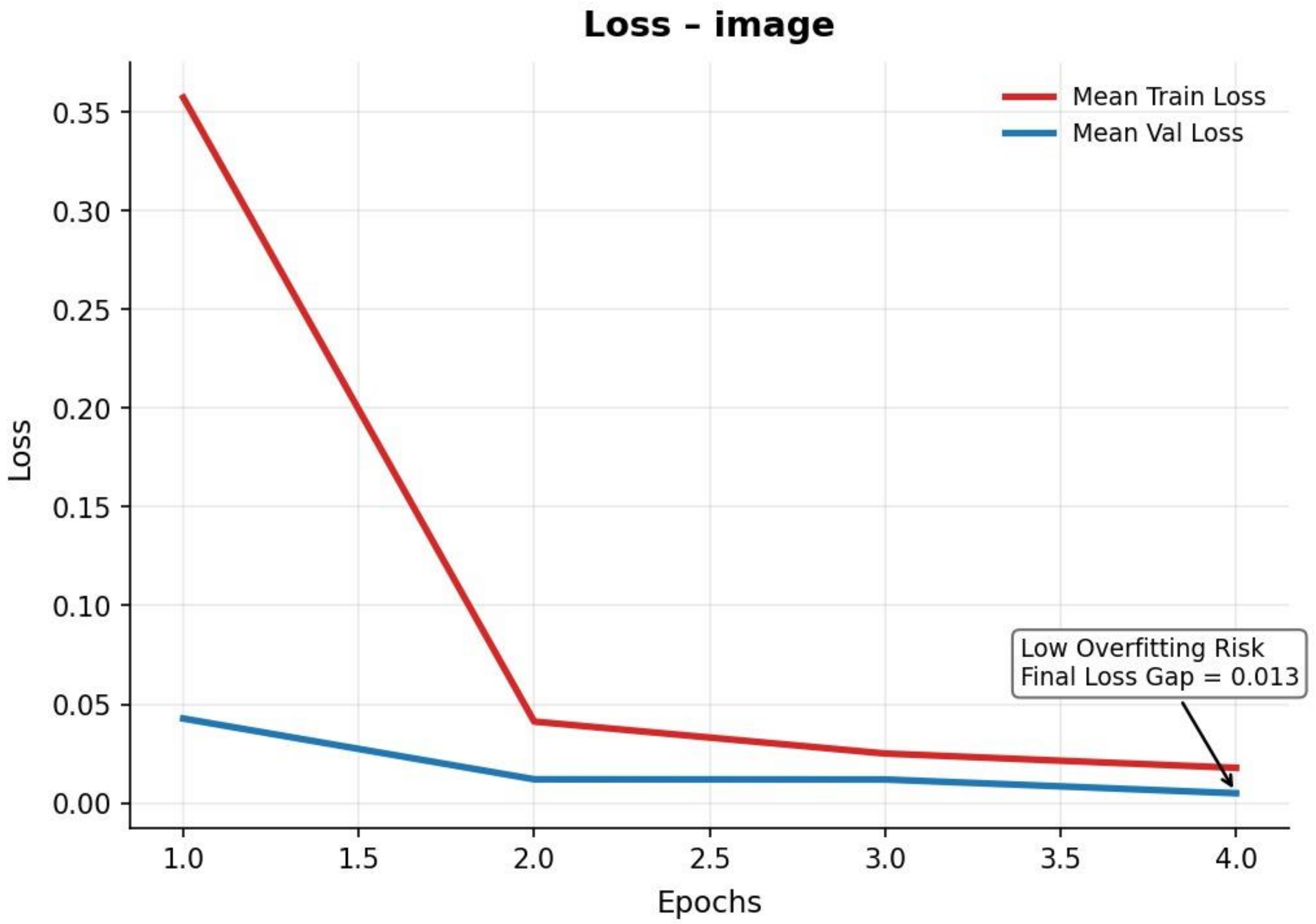
Overfitting Analysis

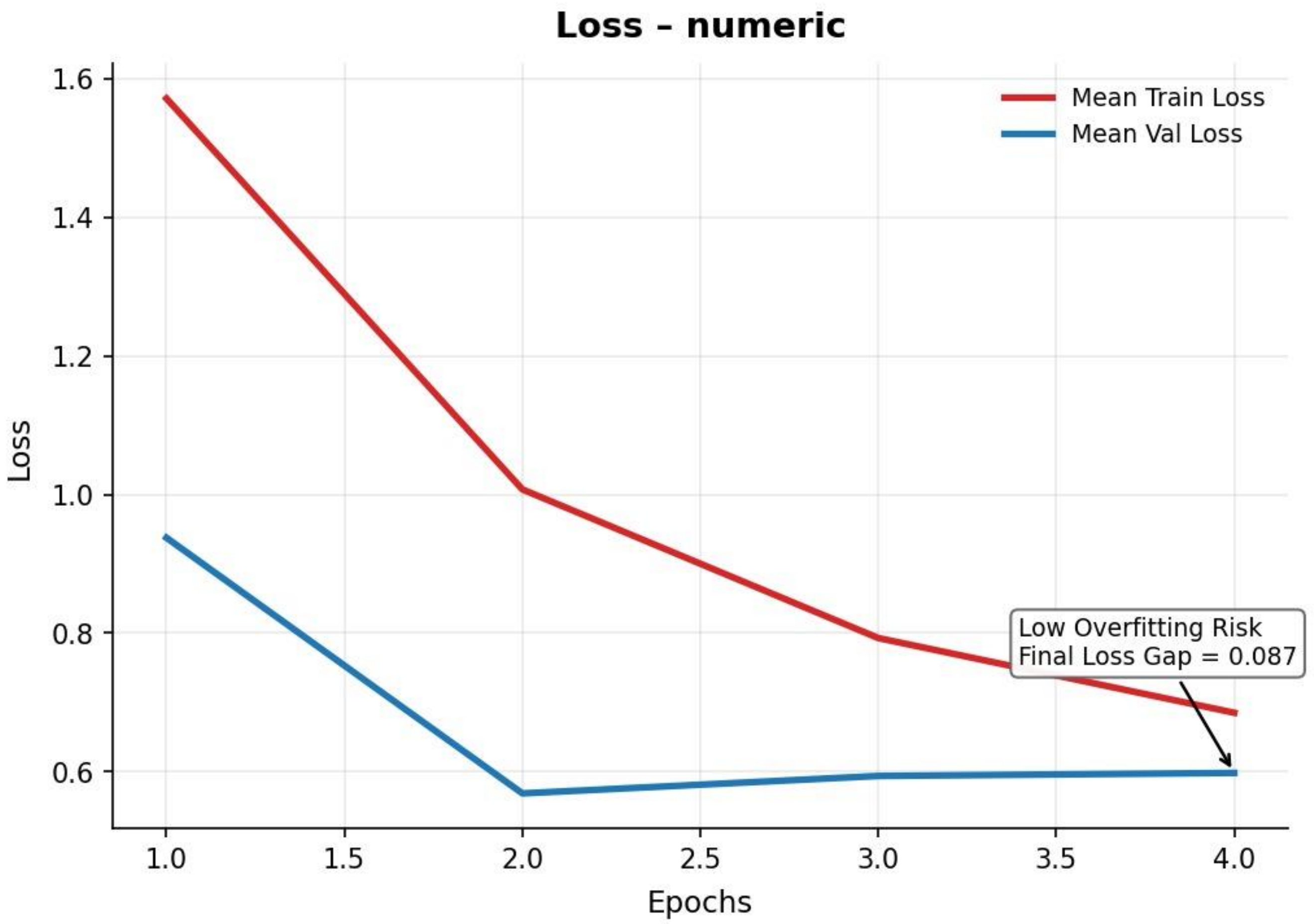




Overfitting Analysis







SOTA Visual Classifiers Comparison

(Skip to Colab-Notebook)

Key Findings

- **Perfect Visual Performance:** Image modality achieves perfect classification (1.0 across all metrics)
- **Visual Dominance:** Any combination including images shows exceptional performance (≥0.996 F1)
- **Multimodal Robustness:** Full multimodal approach achieves near-perfect results (0.996 F1)
- **Optimal Bimodal Combination:** Image + Text combination maintains perfect performance
- **Numerical Limitations:** Hand-crafted numerical features show weakest individual performance

Future Work

- **Advanced Feature Fusion:** Implement cross-modal attention mechanisms to better integrate visual, textual, and numerical information
- **Transformer-based Architecture:** Replace FC layers with transformer blocks for improved feature interactions
- **Learned Numerical Features:** Replace hand-crafted features with deep convolutional feature extractors
- **Robustness:** Enhancing the model's robustness by incorporating a wider range of weather conditions, while introducing noisy images to increase complexity and improve the effectiveness of the training process

Questions?