Transforming Vision, Preserving Oceans: Innovative Vision Transformers with Kolmogorov-Arnold Fourier Embeddings for Coral Classification on Limited Data

Raad Bin Tareaf
r.bintareaf@xu-university.de
XU Exponential University of Applied Sciences
Potsdam, Germany

Alex Maximilian Korga
a.korga@student.xu-university.de
XU Exponential University of Applied Sciences
Potsdam, Germany

Sebastian Wefers
s.wefers@student.xu-university.de
XU Exponential University of Applied Sciences
Potsdam, Germany

Keno Hanken
k.hanken@student.xu-university.de
XU Exponential University of Applied Sciences
Potsdam, Germany

ABSTRACT

The classification of coral health is critical for marine conservation, particularly as coral reefs face increasing threats from climate change and environmental degradation. This study introduces a novel application of Vision Transformers (ViTs) for coral classification, focusing on the integration of Kolmogorov-Arnold Networks (KAN) based Fourier embeddings. This innovative approach leverages the strengths of KAN's learnable functions on edges, which are highly flexible and capable of capturing complex patterns in data. These functions have demonstrated robust performance, particularly in achieving smooth and consistent loss and classification metric curves without signs of overfitting. Compared to convolutional and MLP embeddings, the KAN-Fourier model showed comparable or superior performance across key metrics such as F1 score, precision, and recall, especially on the original dataset. The attention map analysis further revealed that Fourier embeddings effectively capture intricate patterns within coral images, accurately detecting fish, which indicates healthy coral environments, as well as early signs of coral bleaching—patterns that the convolutional and MLP embeddings failed to capture—offering enhanced interpretability. While all models performed well on the test set, the KAN-Fourier variant stood out for its ability to generalize effectively from both original and augmented datasets, suggesting that embedding diversity, particularly the fusion of KAN with Fourier embeddings, can significantly enhance the applicability of ViTs in complex visual tasks like coral health monitoring. These findings have broader implications for environmental science, conservation efforts, and highlight the fascinating emergence of KAN architectures as a novel approach in deep learning.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Deep Learning, Vision Transformer, Kolmogorov-Arnold Networks, Embeddings, Coral Reef Classification, Marine Conservation, Marine Science

1 INTRODUCTION

Coral reefs are among the most diverse and valuable ecosystems on the planet, providing critical habitat for marine species, supporting fisheries, and protecting coastlines from erosion. However, these ecosystems are increasingly threatened by climate change, overfishing, pollution, and other anthropogenic factors. A particularly severe consequence of environmental stress is coral bleaching, a process where corals lose their vibrant colors and essential symbiotic algae, leading to widespread reef degradation and increased mortality rates.

The ability to accurately monitor and assess coral health is crucial for the conservation and restoration of coral reefs. Traditional methods of coral monitoring, which often rely on manual inspection and analysis, are time-consuming and resource-intensive, particularly for large-scale assessments. Advances in deep learning and computer vision offer promising tools to automate and enhance the accuracy of coral health classification, allowing for more efficient and scalable monitoring.

In this paper, we explore the application of Vision Transformers (ViTs) for coral classification, focusing on the innovative integration of Kolmogorov-Arnold Fourier embeddings, alongside more conventional convolutional and linear embeddings. ViTs, which have revolutionized natural language processing and are increasingly being applied to computer vision tasks, offer a powerful architecture that can capture both local and global features in images through self-attention mechanisms.

Our study addresses the challenge of limited data availability, which is a common issue in environmental science applications. We propose and evaluate multiple embedding strategies to enhance the performance of ViTs under such constraints, aiming to improve the generalization and robustness of the models. By applying these methods to a dataset of healthy and bleached coral images, we demonstrate that diverse embedding approaches can significantly impact model performance and interpretability.

The contributions of this paper are fourfold: (1) We introduce a novel application of Kolmogorov-Arnold Fourier embeddings in ViTs for coral classification, (2) We compare the effectiveness of Fourier-based, convolutional, and linear embeddings in this context, (3) We evaluate the impact of data augmentation on model performance, and (4) We provide insights into the interpretability of ViTs

through the analysis of attention maps, offering a transparent view of the decision-making process. Our findings have implications not only for coral reef monitoring but also for the broader application of ViTs in environmental science and other fields where data is limited.

The remainder of this paper is organized as follows: Section 2 reviews related work on coral bleaching, resilience, and the application of deep learning to coral classification. Section 3 describes the dataset used in this study. In Section 4, we detail the methodologies, including the ViT architecture and embedding strategies. Section 5 presents the experimental setup and results. Finally, we conclude with future directions in Section 7.

2 LITERATURE REVIEW

Coral bleaching is a significant and growing concern for marine ecosystems worldwide, driven by various environmental stressors, particularly elevated sea temperatures. This review provides an overview of key studies that have explored the causes, impacts, and broader implications of coral bleaching, setting the stage for the application of deep learning techniques in coral health assessment.

2.1 Mechanisms and Causes of Coral Bleaching

The process of coral bleaching, as detailed by Douglas [2], involves the loss of color in corals due to the breakdown of symbiotic relationships between coral hosts and their dinoflagellate algae, primarily of the genus *Symbiodinium*. Bleaching typically occurs when external stressors, such as elevated sea temperatures, disrupt this symbiosis, leading to the expulsion or degradation of the algal cells, which are crucial for the coral's energy production through photosynthesis. This physiological response is often detrimental, resulting in reduced growth rates and increased mortality among affected corals. The mechanisms underlying these responses are complex and may involve interactions between the algal and animal components of the symbiosis. Variability in bleaching susceptibility is influenced by genetic differences among *Symbiodinium* strains and the acclimatory responses of the coral hosts, though the evolutionary purpose of bleaching remains unclear.

2.2 Resistance and Resilience in Coral Reefs

West and Salm [13] highlight the importance of understanding both the resistance and resilience of coral reefs to bleaching events. Resistance refers to the ability of certain reef areas to withstand bleaching despite exposure to elevated temperatures, while resilience is the capacity of reef communities to recover after bleaching has occurred. These factors are crucial for conservation strategies, as identifying and protecting areas with high resistance and resilience can help mitigate the impact of bleaching on global coral biodiversity. The 1997–1998 El Niño event, one of the most severe on record, underscored the need for such strategies. West and Salm identified environmental factors that could enhance resistance and resilience, such as areas with lower temperature variability, higher water flow, and specific light conditions that reduce stress on corals. These insights are essential for the development of marine protected areas aimed at conserving coral reefs under the threat of ongoing climate change.

2.3 Global Warming and Recurrent Bleaching Events

Hughes et al. [4] examine the recurring nature of mass coral bleaching events, particularly those triggered by record sea temperatures during 2015–2016. This period marked the third global bleaching event since the phenomenon was first observed in the 1980s. Their study, focusing on the Great Barrier Reef, revealed that the geographic extent and severity of bleaching were primarily dictated by sea surface temperatures, with minimal influence from local factors such as water quality or fishing pressure. Notably, previous exposure to bleaching did not confer resistance to subsequent events, indicating that coral reefs remain highly vulnerable to repeated thermal stress. The study concludes that without immediate global action to reduce greenhouse gas emissions, the future of coral reefs is in jeopardy, as they may not be able to survive the increasing frequency and intensity of bleaching events.

2.4 Climate Change and Ocean Acidification

Hoegh-Guldberg et al. [3] provide a comprehensive review of the broader implications of climate change on coral reefs, particularly through the dual threats of global warming and ocean acidification. The authors predict that by the end of the 21st century, atmospheric carbon dioxide levels will exceed 500 parts per million, with global temperatures rising by at least 2°C. Such conditions are unprecedented in the last 420,000 years and are expected to severely compromise the ability of corals to accrete calcium carbonate, the substance that forms their skeletons. As a result, coral reefs are likely to become increasingly rare, leading to less diverse reef ecosystems and the potential collapse of reef structures. The paper emphasizes the urgent need for global action to mitigate climate change, alongside local efforts to improve water quality and reduce overexploitation of marine resources, to prevent the functional collapse of coral reef ecosystems.

2.5 Deep Learning Approaches for Coral Classification

Recent advancements in deep learning have shown promise in the classification of coral health, particularly in distinguishing between healthy and bleached corals. Shihavuddin et al. [10] proposed a novel image classification scheme that combines various texture and color descriptors with classifiers such as support vector machines (SVM) and neural networks. Their method was applied to both single images and composite mosaics of coral reefs, achieving high accuracy in classifying benthic habitats and creating thematic maps of reef ecosystems.

Mahmood et al. [7] explored the use of hybrid feature representations by combining convolutional neural network (CNN) features with hand-crafted features like texture and color descriptors. This approach capitalizes on the complementary strengths of different feature types, resulting in improved classification accuracy on benchmark coral datasets.

Jamil et al. [5] introduced a bag-of-features approach to detect and classify bleached corals. Their method employed various handcrafted descriptors and deep CNNs like AlexNet, GoogLeNet, and ResNet-50 for feature extraction. The proposed technique achieved a classification accuracy of 99.08

Additionally, the Vision Transformer (ViT) model, as discussed by Dosovitskiy et al. [1], represents a shift from traditional CNN-based approaches. ViTs apply transformer architectures directly to image patches, leveraging self-attention mechanisms to capture both local and global features in the image. While ViTs were initially developed for tasks like ImageNet classification, their potential applications to coral reef image classification suggest that they could further improve accuracy and computational efficiency in this domain.

2.6 Relevance to Deep Learning Applications

Understanding the mechanisms, impacts, and variability of coral bleaching is critical for the development of effective deep learning models that can accurately classify coral health. The insights from these studies highlight the importance of factors such as environmental stressors, historical bleaching events, and the resilience of coral species, all of which can be integrated into the data-driven models. By leveraging deep learning, it is possible to enhance the monitoring and conservation of coral reefs, providing a scalable and efficient tool to assess the health of these vital ecosystems and predict their responses to ongoing environmental changes.

Moreover, models solely trained on images to detect healthy and bleached corals can also be invaluable for environmental scientists and practitioners. These models enable rapid, large-scale assessments of coral health, which are crucial for timely decision-making and intervention. By automating the process of coral health monitoring, these models allow for continuous surveillance of coral reefs, reducing the time and resources required for traditional methods. This capability is particularly useful in remote or extensive reef systems where manual monitoring is impractical. Furthermore, the data generated by these models can aid in understanding broader ecological patterns, supporting conservation strategies and policy-making efforts aimed at mitigating the effects of climate change and other environmental threats on coral reefs.

3 DATA

The dataset used in this study is the "Healthy and Bleached Corals Image Classification" dataset, sourced from Kaggle [12]. This dataset is specifically curated to assist in the classification of healthy versus bleached corals, providing a valuable resource for researchers and machine learning practitioners. It consists of a total of 923 images, which have been categorized into two distinct classes: healthy corals (438 images) and bleached corals (485 images).

The images were collected from Flickr using the Flickr API. The primary objective of this dataset is to support the development and evaluation of classification models that can accurately differentiate between healthy and bleached corals. Such models are crucial for monitoring the health of coral reefs and contributing to their conservation.

As illustrated in Figure 1, the images on the left show two examples of healthy corals, characterized by their vibrant colors and robust structure. In contrast, the images on the right depict bleached corals, which are visibly pale and lack the vibrant coloration, indicative of stress and potential decline in health. Healthy coral reefs are characterized by vibrant marine life, including fish and other creatures, which provide informative features for the model to learn and focus on. Note that the bleaching event of the coral in the image on the bottom left corner had just begun and presented obvious challenges for the classifiers.

By creating effective classification models, this dataset aids in enhancing our understanding of coral reef health, thereby contributing to efforts aimed at preserving these vital marine ecosystems.

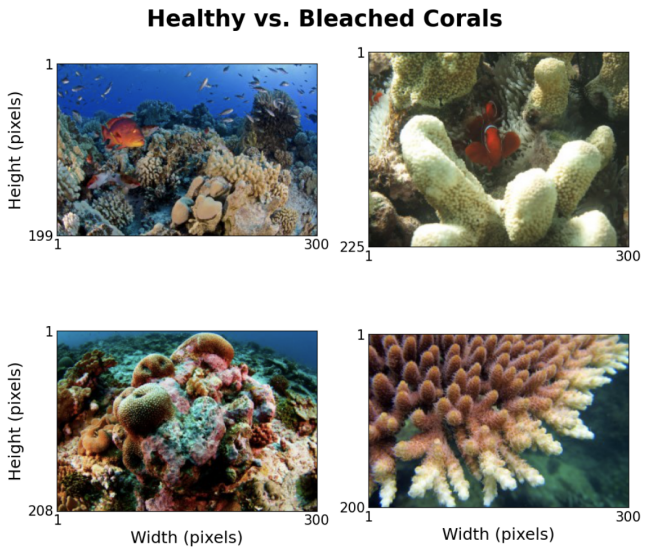

Figure 1: Healthy vs. Bleached Corals. The left column shows examples of healthy corals, while the right column shows images of bleached corals.

3.1 Data Cleaning

To ensure the dataset's consistency, images with dimensions exceeding 300 pixels were excluded, leaving a final dataset of appropriately sized images. The dataset was then split into training, validation, and test sets with a respective ratio of 70:20:20. We ensured that the RGB color distributions across these splits were statistically similar, which is essential for maintaining the representativeness of each subset.

These consistent RGB distributions across the training, validation, and test sets, as displayed in Figure 2, help ensure that the model is trained on representative samples, which is critical for achieving reliable performance during testing. In the figure, each bar represents the mean pixel intensity for one of the RGB channels (Red, Green, Blue) across all images in the respective dataset. The height of the bar indicates this average intensity, giving a sense of the overall color balance in the dataset.

The error bars extending above each bar represent the standard deviation, which indicates the amount of variation or dispersion from the mean for the pixel intensities in each channel. A smaller standard deviation suggests that the pixel intensities are more tightly clustered around the mean, while a larger standard deviation suggests greater variability.

By ensuring that both the mean and standard deviation are consistent across the training, validation, and test sets, we can help

prevent biases in the model's learning process, promoting better generalization to unseen data.

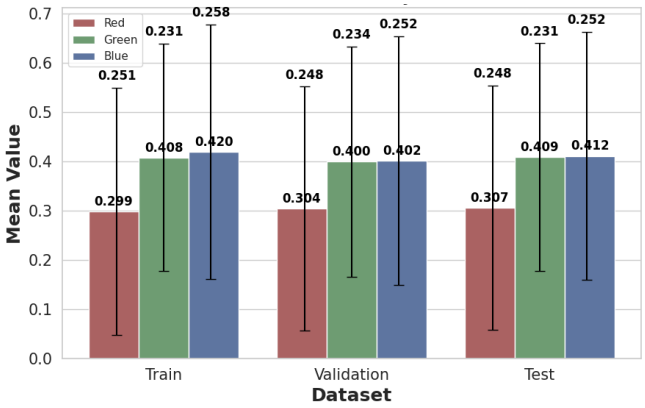

Figure 2: RGB Mean and Standard Deviation Distribution across Train, Validation, and Test Sets.

3.2 Data Preprocessing

Data preprocessing plays a crucial role in ensuring the quality and consistency of input data for deep learning models. In this study, the image dataset underwent several preprocessing steps to standardize the dimensions and ensure the dataset's integrity. Initially, images with dimensions greater than 300 pixels in either width or height were removed from the dataset. This decision was based on the analysis of image dimensions, as shown in Figure 3. The remaining images, which had minimum dimensions of 150x134 pixels and a maximum of 300x300 pixels, were then subjected to further processing.

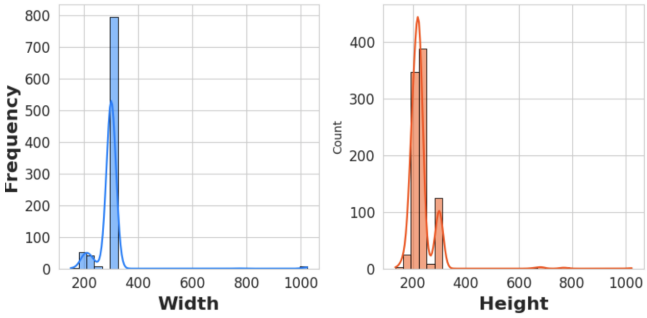

Figure 3: Pixel Height and Width Distributions of the Dataset.

The preprocessing pipeline involved the following key steps:

(1) **Padding and Resizing**: Images smaller than the target size of 300x300 pixels were padded symmetrically to maintain the aspect ratio before resizing. The padding mode used was 'reflect' or 'edge' to avoid introducing artifacts that could mislead the model during training.
(2) **Scaling and Transformation**: After padding, the images were resized to a uniform size of 300x300 pixels. Subsequently, the pixel values were scaled and transformed into tensors for compatibility with the PyTorch framework. For the MLP and Convolutional embedding-based models, the pixel values were scaled to the range $[0, 1]$. However, for the Fourier variant, the pixel values were scaled to the range $[-\pi, +\pi]$ to better align with the periodic nature of the sine and cosine transformations.
(3) **Train-Validation-Test Split**: The dataset was split based on 60:20:20 ratio and in a stratified fashion to ensure a consistent target variable distribution across all sets.

Additionally, the label distributions across these splits were relatively balanced, with 333 healthy and 293 bleached corals in the training set, 48 healthy and 42 bleached corals in the validation set, and 95 healthy and 84 bleached corals in the test set.

3.3 Data Augmentation Techniques

To enhance the diversity and robustness of our training dataset, we employ data augmentation techniques focused on adjusting image brightness, contrast, and saturation, along with applying random flips and resizing. Given the variability in underwater conditions, these adjustments are crucial for simulating different lighting and water clarity scenarios. We first compute the statistical distribution parameters (mean, standard deviation, minimum, and maximum values) for brightness, contrast, and saturation across the dataset. However, instead of sampling directly from these natural distributions, we use uniform sampling within the observed range. This approach introduces a broader variety of transformations, ensuring that the model encounters a wider spectrum of possible visual appearances, including more extreme cases that may be underrepresented in the actual data. Additionally, random horizontal and vertical flips, with probabilities of 0.5 and 0.2 respectively, are applied to account for different coral orientations. Finally, images are padded and resized to maintain consistent input dimensions. This combination of techniques, particularly the use of uniform sampling, helps the model generalize better to diverse and unseen data conditions.

4 METHODOLOGY

In this section, we outline the methodology employed in this study, focusing on the design and implementation of the Vision Transformer (ViT) architecture for coral classification. This includes a detailed description of the core components of the ViT model, inspired by the pioneering work of Dosovitskiy et al. [1], and rooted in the self-attention mechanisms introduced by Vaswani et al. [11]. We also explore the theoretical foundations of the Kolmogorov-Arnold networks and their application in Fourier-based embedding layers, which play a crucial role in enhancing the model's ability to capture complex patterns in the data.

Furthermore, we discuss the diverse embedding strategies implemented in our model, including Fourier-based, linear, and convolutional embeddings. These techniques are designed to leverage the strengths of each embedding type, adapting the ViT architecture to the specific challenges posed by our dataset. Finally, we describe the data augmentation techniques used to improve the robustness and generalization of the models during training.

This methodology section provides a comprehensive overview of the architectural choices, theoretical foundations, and practical

considerations that underpin our approach to coral classification using Vision Transformers.

4.1 Vision Transformer Architecture

The Vision Transformer (ViT) architecture, as introduced by Dosovitskiy et al. [1], represents a significant shift from traditional convolutional neural networks (CNNs) by directly applying transformer models to image patches. This approach leverages the power of self-attention mechanisms, which were originally introduced by Vaswani et al. in their seminal work on transformers [11]. These self-attention mechanisms have become the foundation for many state-of-the-art models in both natural language processing and computer vision.

In our implementation, we adopt a similar architecture with several key components, which are outlined below:

Patch Embedding: The first step in the ViT architecture is to divide an input image into smaller patches. Each patch is then linearly embedded into a vector that represents its features. In our model, the `PatchEmbedding` class handles this process, allowing for three types of embeddings—`conv`, `linear`, and `fourier`—depending on the variant being tested. The class projects the input image patches into an embedding space of a specified dimension, which is then used as input to the transformer blocks.

Transformer Blocks: Following the patch embedding, the embedded patches are passed through a series of transformer blocks. Each block consists of multi-head self-attention mechanisms, as introduced by Vaswani et al. [11], followed by a feed-forward neural network (MLP). Additionally, each transformer block incorporates layer normalization and residual connections around both the attention mechanism and the MLP, which helps stabilize training and improve performance. These blocks are designed to capture long-range dependencies between the patches, enabling the model to learn complex patterns in the image. Our implementation uses the `TransformerBlock` class to define these operations.

Classification Head: After the transformer blocks, the output corresponding to a learnable classification token is passed through an MLP head, which is responsible for making the final classification decision. This head uses a layer normalization followed by a linear transformation.

Training Setup: The model is trained using binary cross-entropy loss with logits (`BCEWithLogitsLoss`), and performance is evaluated using metrics such as accuracy, micro F1 score, micro precision, and micro recall. The optimizer used is Adam, with a learning rate specified during model initialization.

This architecture is tested with different embedding types, including convolutional, linear, and Fourier-based embeddings, to evaluate their performance in the context of coral classification. The model's ability to capture both local and global patterns is facilitated by the self-attention mechanism, which can process long-range dependencies more effectively than traditional CNNs.

4.2 Theoretical Background of Kolmogorov-Arnold Networks

The Kolmogorov-Arnold Representation Theorem underpins a fundamental concept in function analysis, asserting that any multivariate continuous function can be represented as a superposition of continuous functions of one variable. Formally, the theorem is expressed as:

$$f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \phi_q \left(\sum_{p=1}^{n} \psi_{pq}(x_p) \right)$$

where ϕ_q and ψ_{pq} are continuous functions of one variable. This theorem provides a powerful theoretical framework, suggesting that complex dependencies in high-dimensional data can be effectively decomposed into simpler, univariate functions.

Building upon this foundational concept, the Kolmogorov-Arnold Networks (KAN) were introduced as an innovative neural architecture inspired by the theorem. KANs offer a promising alternative to traditional Multi-Layer Perceptrons (MLPs) by replacing fixed activation functions with learnable, univariate functions on edges (or weights), as described by Liu et al. [6]. This seemingly simple change allows KANs to outperform MLPs in terms of accuracy, interpretability, and mathematical rigor.

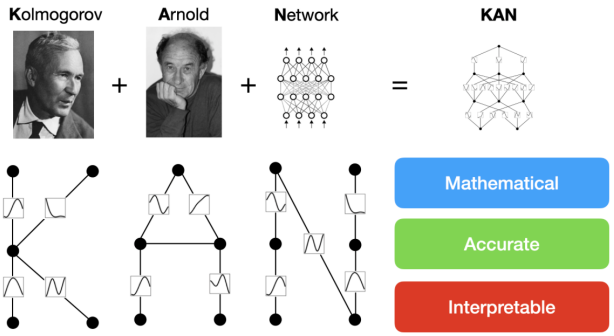

Figure 4: Kolmogorov-Arnold Network (KAN) illustration: a visual representation combining the contributions of Andrey Kolmogorov and Vladimir Arnold with modern neural network architecture. The image emphasizes the mathematical foundation, accuracy, and interpretability of KANs. Image adapted from the *"Awesome KAN"* GitHub repository [8].

As illustrated in Figure 4, the KAN architecture can be visualized as a hybrid model that blends traditional neural networks with advanced mathematical theories. The image combines the faces of Andrey Kolmogorov and Vladimir Arnold with a visual representation of MLPs, symbolizing the fusion of mathematical theory with neural network design. The three bullet points—mathematical, accurate, and interpretable—highlight key advantages of KANs:

- **Mathematical:** KANs are deeply rooted in the Kolmogorov-Arnold theorem, providing a mathematically rigorous foundation that enhances their theoretical soundness.
- **Accurate:** KANs have demonstrated superior accuracy in various tasks, including data fitting and solving partial differential equations (PDEs), often outperforming traditional MLPs.
- **Interpretable:** Unlike traditional neural networks, KANs offer a level of interpretability that allows for intuitive visualization and interaction, making them valuable tools for scientific discovery.

The visual and conceptual emphasis on these three attributes underscores the potential of KANs to revolutionize the way we approach neural network design, particularly in fields that require both precision and transparency.

Moving from these foundational concepts of function decomposition, we apply the principles derived from Kolmogorov-Arnold networks to modern neural architectures, particularly focusing on embedding layers. The application of KAN principles in our Fourier-based embedding layer demonstrates the practical benefits of this theoretical framework in enhancing model performance and interpretability in complex tasks such as image classification.

4.3 Application in Fourier-based Embedding Layer

Building on the principles of the Kolmogorov-Arnold theorem, our Fourier-based embedding layer, termed the Naive Fourier Komogorov-Arnold Layer (KAN Layer), employs trigonometric functions to achieve a similar decomposition:

$$y = \sum_{k=1}^{\texttt{gridsize}} \left(\cos(kx) \cdot \text{coeff}_{\cos + \sin(kx) \cdot \text{coeff}_{\sin}} \right)$$

In this framework, k indexes the grid points, resembling the summation over univariate functions in the theorem. The use of cosine and sine functions—fundamental to Fourier analysis—serves to approximate the continuous functions ϕ and ψ, thereby embedding the input vector x into a higher-dimensional, analyzable space.

By integrating Fourier analysis, we extend the theorem's conceptual approach, adapting its core idea to the domain of neural networks and leveraging it for complex pattern recognition tasks such as image classification.

Building on the Fourier transformation's role in approximating complex function interactions, we next explore several embedding strategies that further exploit this mathematical framework within a Vision Transformer architecture.

4.4 Embedding Layers

The use of diverse embedding techniques is crucial for adapting the Vision Transformer to specific challenges posed by our dataset and task. Below, we detail each embedding strategy implemented in our model, discussing their unique benefits and roles in enhancing model performance and interpretability.

4.4.1 Fourier-based Embedding Layer

The Fourier-based embedding layer, specifically the Naive Fourier Komogorov-Arnold Layer (KAN Layer), transforms input features using trigonometric transformations. This implementation is inspired by and adapted from the FourierKAN repository [9], which utilizes Fourier transformations to model input features. Unlike traditional linear layers followed by explicit non-linear activations, the KAN-Fourier layer introduces non-linearity inherently through the use of learnable sine and cosine functions, drawing its inspiration from Kolmogorov-Arnold Networks (KAN). However, instead of using spline coefficients, this layer employs 1D Fourier coefficients, which are denser and more globally effective compared to the local nature of splines. This makes Fourier coefficients easier to optimize, as they inherently offer a more numerically bounded, periodic function that mitigates the risk of values going out of grid bounds.

The layer performs the following operations:

(1) Each input vector $x \in \mathbb{R}^{\texttt{inputdim}}$ is transformed into a higher-dimensional space using the following mapping:

$$y = \sum_{k=1}^{\texttt{gridsize}} (\cos(kx) \cdot \text{coeff}_{\cos} + \sin(kx) \cdot \text{coeff}_{\sin}) \tag{1}$$

where k indexes the grid points, and $\text{coeff}_{\cos}, \text{coeff}_{\sin} \in \mathbb{R}^{\texttt{outdim} \times \texttt{inputdim} \times \texttt{gridsize}}$ are the Fourier coefficients, learned parameters of the model. The sine and cosine components are summed to interpolate the data, effectively creating a complex signal from these simpler sinusoidal functions.

(2) The coefficients are normalized by a factor depending on the `smooth_initialization` parameter:

$$\text{norm\_factor} = \begin{cases} \text{range}(1, \texttt{gridsize} + 1)^2 & \text{if smooth\_initialization is true} \\ \sqrt{\texttt{gridsize}} & \text{otherwise} \end{cases} \tag{2}$$

This normalization ensures that the output coordinates have unit variance when the input coordinates have unit variance, independent of the grid size.

(3) Optionally, a bias term $\text{bias} \in \mathbb{R}^{\texttt{outdim}}$ can be added to the output.

One of the key advantages of using Fourier coefficients over splines, as highlighted in the FourierKAN repository, is that Fourier functions are periodic and therefore more numerically stable. This periodicity helps to prevent issues that arise when values exceed predefined grid bounds, offering a more robust solution for modeling complex functions. Additionally, after achieving convergence, the Fourier-based functions can be replaced with spline approximations for faster evaluation, which provides nearly identical results with improved computational efficiency.

4.4.2 Linear Embedding Layer

The linear embedding layer applies a linear transformation to each flattened input patch, transforming it into a token embedding suitable for sequential processing. For a patch of dimension `input_dim` and an output embedding of dimension `embed_size`, the transformation is defined by:

$$y = Wx + b$$

where $x \in \mathbb{R}^{\texttt{input\_dim}}$ represents the flattened input patch. Here, $W \in \mathbb{R}^{\texttt{embed\_size} \times \texttt{input\_dim}}$ and $b \in \mathbb{R}^{\texttt{embed\_size}}$ are learnable parameters. This embedding is particularly effective in maintaining a balance between computational efficiency and representational power, making it suitable for scenarios with dense input features.

4.4.3 Convolutional Embedding Layer

The convolutional embedding layer utilizes a convolutional operation to process input images directly into embedded patches. This method involves using a convolutional layer with a kernel size and stride equal to the patch size, effectively partitioning the image into non-overlapping patches and directly extracting features. The transformation can be represented as:

$$x' = \text{Conv2d}(x; W, b)$$

where x is the input image, W is the convolutional kernel, and b is the bias. The output x' represents the sequence of patch embeddings.

4.5 Evaluation Metrics

In this study, we employ micro precision, recall, and F1-score as the primary evaluation metrics to assess the performance of our models. The choice of micro-averaged metrics is particularly important in scenarios where there is an imbalance in the data distribution across classes. Micro averaging aggregates the contributions of all classes to compute the average metric, ensuring that each instance, regardless of its class, is given equal weight. This approach is beneficial when the dataset contains a disproportionate number of examples across different classes, as it provides a more balanced view of the model's performance.

The formulas for these metrics are as follows:

- **Micro Precision**: Micro precision is calculated as the total number of true positives across all classes divided by the total number of predicted positives across all classes:

$$\text{Micro Precision} = \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} (\text{TP}_i + \text{FP}_i)}$$

  where TP_i and FP_i are the true positives and false positives for class i, respectively, and n is the total number of classes.
- **Micro Recall**: Micro recall is calculated as the total number of true positives across all classes divided by the total number of actual positives across all classes:

$$\text{Micro Recall} = \frac{\sum_{i=1}^{n} \text{TP}_i}{\sum_{i=1}^{n} (\text{TP}_i + \text{FN}_i)}$$

  where FN_i are the false negatives for class i.
- **Micro F1-Score**: The micro F1-score is the harmonic mean of micro precision and micro recall:

$$\text{Micro F1} = 2 \cdot \frac{\text{Micro Precision} \cdot \text{Micro Recall}}{\text{Micro Precision} + \text{Micro Recall}}$$

- **Accuracy**: Accuracy is calculated as the total number of correct predictions divided by the total number of predictions:

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \text{TP}_i + \text{TN}_i}{\sum_{i=1}^{n} (\text{TP}_i + \text{TN}_i + \text{FP}_i + \text{FN}_i)}$$

  where TN_i are the true negatives for class i.

While micro-averaged metrics are essential for handling imbalanced datasets, accuracy is also retained as an evaluation metric. Accuracy, which measures the ratio of correctly predicted instances to the total number of instances, remains a widely recognized and interpretable metric. It offers a straightforward evaluation of overall model performance and is particularly useful when the dataset is relatively balanced. Even though accuracy can be misleading in the presence of class imbalance, it still provides valuable insights when interpreted alongside micro precision, recall, and F1-score, offering a more comprehensive evaluation of the model's effectiveness.

4.6 Binary Cross-Entropy With Logits Loss

Binary Cross-Entropy With Logits Loss (BCEWithLogitsLoss) is commonly used for binary classification tasks. This loss function combines a sigmoid layer and the binary cross-entropy loss in one single class. Given the predicted logits z and the target label y, the BCEWithLogitsLoss is defined as:

$$\text{BCEWithLogitsLoss}(z, y) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i)) \right]$$

Where:

- N is the number of samples.
- y_i is the binary ground truth label for the i-th sample.
- z_i is the logit (raw model output) for the i-th sample.
- $\sigma(z_i)$ is the sigmoid function applied to the logit z_i, defined as:

$$\sigma(z_i) = \frac{1}{1 + e^{-z_i}}$$

This loss function is particularly useful as it directly operates on logits, avoiding potential numerical instability issues that might arise from directly applying the sigmoid function before computing the binary cross-entropy.

4.7 Adam Optimizer

Adam (short for Adaptive Moment Estimation) is an optimization algorithm that computes adaptive learning rates for each parameter. It is widely used due to its efficiency and effectiveness in training deep learning models. The optmimizer efficiently combines the advantages of two other extensions of stochastic gradient descent: Root Mean Square Propagation (RMSProp) and Momentum. It is known for its ability to handle sparse gradients on noisy problems and is often the default choice for training neural networks.

5 EXPERIMENTS & RESULTS

In this section, we present a comprehensive evaluation of three variants of vision transformers—Kolmogorov-Arnold Fourier, Convolutional, and MLP—applied to coral classification. The experiments are designed to assess the effectiveness of these models across different data settings, including training on both original and augmented datasets. We detail the setup of these experiments, including data preparation, model architectures, and training procedures.

Subsequently, we perform a comparative analysis of the models' performance, examining accuracy next to micro F1, micro precision and micro recall as well as the losses across training and validation sets. To provide further insight into the models' decision-making processes, we also analyze their attention maps, highlighting the explainability and interpretability of each approach. This section aims to explain the strengths and weaknesses of each type of image converter to provide a comprehensive understanding of their capabilities in the context of coral classification.

5.1 Experiment Setup

In this section, we describe the experimental setup for training three variants of vision transformers on the coral classification dataset. The chosen variants—Kolmogorov-Arnold Fourier, Convolutional, and MLP-based embeddings—are designed to explore different ways of representing image data before applying transformer blocks. This comparison is motivated by the distinct properties of each embedding type, which influences model performance in unique ways.

5.1.1 Embedding Techniques for Vision Transformers

The three vision transformer architectures evaluated in this study differ primarily in the way they embed the input image patches:

- **Kolmogorov-Arnold Fourier Embedding:** In this variant, each input image is divided into non-overlapping patches, which are then flattened into vectors. The KAN-Fourier embedding applies a series of sine and cosine transformations to each element of these flattened vectors, with the frequency of these transformations controlled by the 'grid-size' parameter (sin and cos pairs). The resulting waveforms, each modulated by learnable Fourier coefficients, are summed to interpolate the data, effectively creating a complex signal by summing multiple simpler sinusoidal signals. This process allows the embedding to capture periodicities and global patterns in the image, making it particularly effective for identifying regular textures or repeating structures in coral images, as it captures a wide range of frequency components.
- **Convolutional Embedding:** This variant uses a convolutional layer to process each patch. The input image is divided into patches, and a convolutional layer is applied to these patches to extract local features. The convolutional operation preserves spatial hierarchies and captures localized patterns, such as edges and textures, which are crucial for recognizing detailed coral structures. The extracted features are then reshaped into a sequence suitable for the transformer model.
- **MLP Embedding:** In the MLP-based variant, each patch is treated as a flattened vector, and a fully connected (linear) layer is applied to these vectors. This approach allows the model to learn flexible and potentially complex representations from the raw pixel values of the patches. The MLP embedding is advantageous when dealing with diverse and irregular patterns in coral images, as it provides the model with the capacity to learn non-linear combinations of the input features.

Each model processes the embedded patch sequences through a series of transformer blocks, which consist of multi-head self-attention mechanisms and feed-forward neural networks (MLPs). The addition of a learnable classification token and positional embeddings ensures that the model can capture both the spatial structure and overall context of the input image, enabling effective classification of coral health states.

5.1.2 Hyperparameters

The hyperparameters used for the different models across both original and augmented data are summarized in Table 1.

For the KAN-Fourier model, a consistent configuration was maintained for both datasets, with the key difference being the scaling type, set to π-scale for better alignment with the Fourier embedding. The model utilized a grid size of 8, and the augmented data training introduced the start of augmentation from epoch 10.

In the Convolutional model, while the architecture remained similar for both original and augmented data, with a patch size of 16 and embedding size of 66, padding modes were adjusted dynamically during augmentation (reflect/edge). Similarly, the MLP model shared much of its configuration with the KAN-Fourier model, though the scaling type was set to min-max, with augmentation also starting from epoch 10.

All models were trained for 30 epochs using the Adam optimizer. Batch sizes ranged from 10 to 12, depending on the model. Mixed precision on GPU was applied to optimize computational performance while maintaining accuracy, and dropout rates were adjusted based on the dataset to prevent overfitting.

Legend:

* Padding mode switches between edge and reflect for the augmented data.

† Indicates values used specifically for models trained on augmented data.

The models were trained using the Adam optimizer, with batch sizes ranging from 10 to 12 depending on the specific model. Mixed precision on GPU was employed across all experiments, ensuring efficient use of resources while maintaining model performance.

5.1.3 Hardware and Computational Resources

The experiments in this study were conducted on a machine equipped with the following hardware specifications:

- **CPU:** 8-core processor
- **RAM:** 30 GiB
- **GPU:** NVIDIA Quadro P6000 with 24 GiB of GPU memory

The availability of 24 GiB of GPU memory allowed for efficient training of deep learning models, particularly when working with large image datasets and complex architectures such as Vision Transformers. However, due to the computational demands of Kolmogorov-Arnold networks, further exploration of the parameter space was limited by the available hardware resources. Future work would benefit from more powerful GPUs to enhance model performance and reduce training times.

5.2 Training on Original Data

In this section, we analyze the performance of the three vision transformer variants—Fourier-KAN, Convolutional, and MLP—when trained on the original coral classification dataset. The training process was monitored over 30 epochs, and the models' performance was evaluated based on their loss curves and classification metrics, including F1 score, precision, and recall.

5.2.1 Fourier-KAN Embedding

As illustrated in Figure 5, the Fourier-KAN model demonstrates strong performance across the board, with both the loss curves and the metric curves exhibiting consistent and desirable trends. The loss curves converge nicely, following a smooth downward trajectory throughout the 30 epochs, indicating effective learning and minimal signs of overfitting.

Furthermore, the F1 score, precision, and recall metrics all show a steady upward trend, with curves that not only converge but also start at relatively high values compared to other embedding variants. This early strong performance underscores the inherent strength of the Fourier embedding in capturing the underlying data structure from the outset.

The deltas between the training and validation metrics are minimal, with a loss delta of -0.02, an F1 delta of 0.0, a precision delta of 0.02, and a recall delta of 0.04. These small deltas indicate that the

Table 1: Embedding Parameters and Hyperparameters for Models Trained on Original and Augmented Data

Parameter	KAN-Fourier	Conv	MLP
Batch Size	10	12	12
Scaling Type	pi-scale	min-max	min-max
Padding Mode	edge / edge, reflect*	edge / edge, reflect*	edge / edge, reflect*
Patch Size	10	16	10
Embedding Size	64 / 68†	66 / 68†	64 / 68†
Depth	2 / 3†	2 / 3†	2 / 3†
Heads	2 / 4†	2 / 4†	2 / 4†
MLP Dimension	128	128	128
Dropout	0.1 / 0.2†	0.2 / 0.2	0.1 / 0.2†
Learning Rate	1e-5	1e-4	1e-4
Grid Size	8	None	None
Add Bias	True	None	None
Smooth Initialization	False	None	None
Epochs	30	30	30
Start Augmentation Epoch	None / 10†	None / 10†	None / 10†

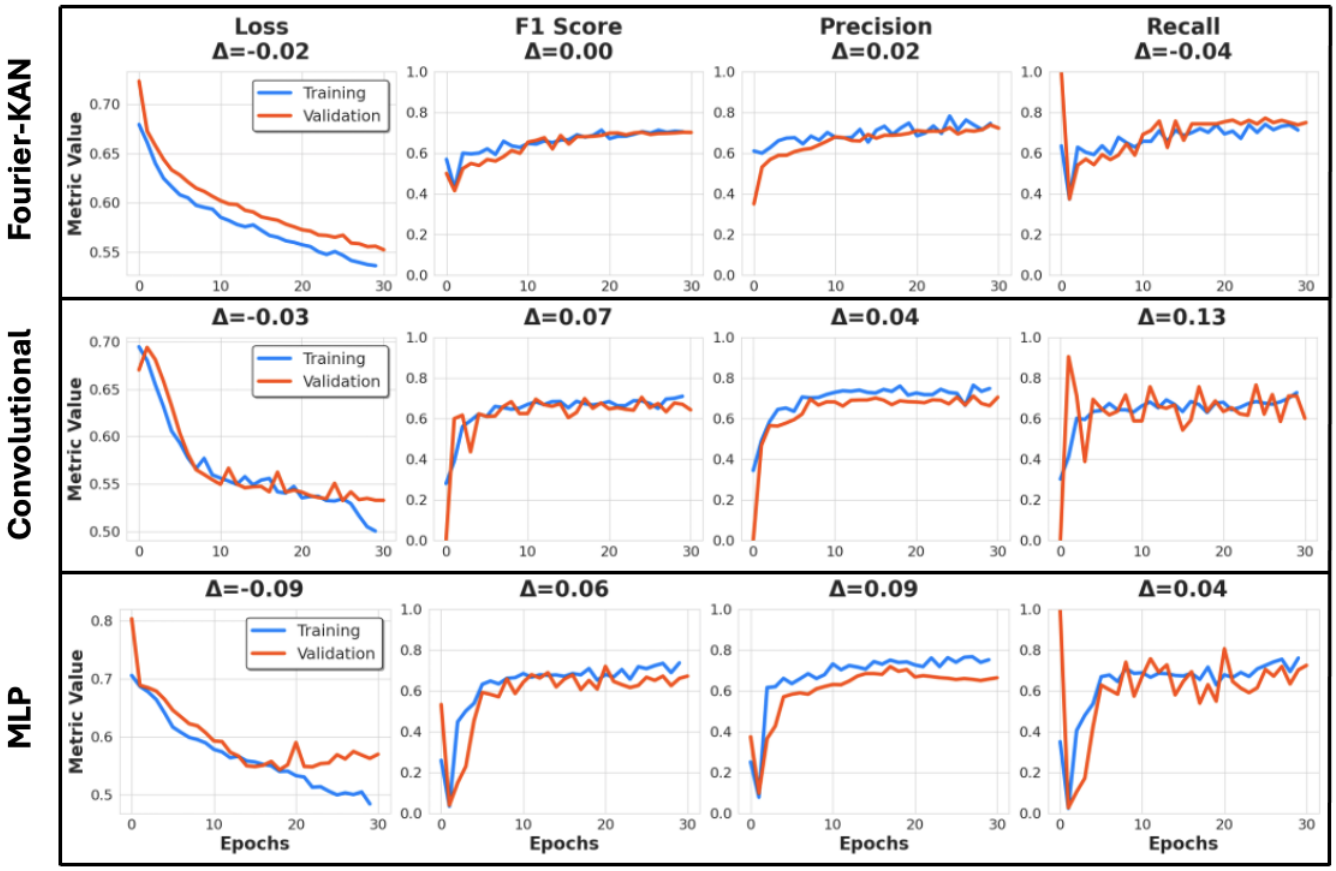

Figure 5: Comparison of Training and Validation Performance Across Different Embedding Variants on Original Data.

Fourier-KAN model generalizes well to unseen data, maintaining robust performance across different evaluation metrics without significant overfitting.

5.2.2 Convolutional Embedding

In contrast to the Fourier-KAN model, the Convolutional variant, as depicted in Figure 5, demonstrates better convergence. The loss curves between the training and validation sets converge more closely, reflecting a more balanced learning process. The F1 score delta of 0.07, precision delta of 0.04, and recall delta of 0.13 indicate that the model generalizes reasonably well, with minimal overfitting. The classification metrics' curves flow in parallel and exhibit a consistent upward trend, which is indicative of the model's steady improvement over the epochs.

5.2.3 MLP Embedding

The MLP-based model, as shown in Figure 5, also shows converging loss curves, though there is a slight divergence observed between the 20th and 30th epochs. Despite this, the model maintains relatively low deltas for the classification metrics, with an F1 score delta of 0.06, precision delta of 0.09, and recall delta of 0.04. These small deltas suggest that the model manages to balance learning and generalization effectively. Like the Convolutional model, the MLP variant's classification metrics demonstrate parallel and upward trends, reflecting a consistent and stable learning process.

Overall, Figure 5 reveals that all three models—Fourier-KAN, Convolutional, and MLP—exhibit strong performance with consistent convergence and generalization across the original dataset. The Fourier-KAN model, in particular, stands out for its strong initial performance and minimal deltas between training and validation metrics, indicating robust learning and effective generalization with little to no overfitting.

The Convolutional model also demonstrates commendable convergence, with close alignment between training and validation loss curves and moderate deltas in classification metrics, suggesting a balanced learning process. Similarly, the MLP model maintains a steady learning trajectory with slightly larger metric deltas but still achieves effective generalization.

In conclusion, while all three models perform well, the Fourier-KAN model's early strength and smooth convergence, combined with minimal discrepancies between training and validation metrics, suggest it may have an edge in leveraging the data structure effectively, although the Convolutional and MLP models also show solid generalization capabilities.

5.3 Training on Augmented Data

In this section, we explore the performance of the three vision transformer variants—Fourier-KAN, Convolutional, and MLP—when trained on the augmented coral classification dataset. Similar to the original data analysis, the models were monitored over 30 epochs, with evaluations based on loss curves and key classification metrics, including F1 score, precision, and recall.

5.3.1 Fourier-KAN Embedding

As depicted in Figure 6, the Fourier-KAN model continues to demonstrate robust performance when trained on the augmented data. The loss curves maintain a smooth downward trajectory, closely mirroring those observed in the original data, with a final loss delta of 0, indicating consistent learning with minimal overfitting.

The classification metrics, including F1 score, precision, and recall, show similarly strong trends. The F1 score exhibits a steady upward trend, converging with a final delta of 0.04, while precision, though slightly bumpier, converges well with a delta of 0.06. However, recall shows a slightly larger gap towards the last epochs (20-30), with a delta of 0.12, yet it still follows a generally converging trend.

Overall, the Fourier-KAN model exhibits a clear ability to leverage the augmented data effectively, maintaining strong performance across all metrics, though the recall metric shows some divergence in later epochs.

5.3.2 Convolutional Embedding

The Convolutional model, as shown in Figure 6, also performs well on the augmented data, although with some noticeable differences compared to the original dataset. The loss curve follows a general downward trend, but begins to show increased bumpiness after the 10th epoch, tending to diverge slightly around the 25th epoch, with a final loss delta of -0.05.

The F1 score shows a strong upward trend, closely aligning towards convergence with a final delta of 0, suggesting robust generalization. Precision follows a similar pattern, though with a slightly larger gap towards the end, resulting in a delta of 0.05. The recall metric displays a rather bumpy pattern during the first half of the epochs, but it aligns and converges well in the second half, ending with a delta of 0.03.

Despite some bumps in the loss curve, the Convolutional model continues to exhibit solid performance, with particularly strong convergence in the classification metrics, indicating effective learning from the augmented data.

5.3.3 MLP Embedding

The MLP-based model, as illustrated in Figure 6, shows a similar overall pattern to the Convolutional model, with the loss curve converging well but becoming bumpier towards the end, resulting in a loss delta of -0.04.

The F1 score demonstrates a nicely overlaying upward trend, albeit with some bumpiness at the beginning, and converges with a final delta of 0.03. Precision also exhibits a strong overlaying converging pattern, ending with a delta of 0.03. The recall metric, while somewhat bumpy across all epochs, ultimately converges with a delta of 0.04.

In summary, the MLP model manages to maintain a consistent learning trajectory, effectively generalizing across the augmented data, though it does show slightly more variability compared to the other models.

5.3.4 Comparison and Conclusion

The analysis of the augmented data reveals that all three models—Fourier-KAN, Convolutional, and MLP—continue to perform well, with consistent convergence and generalization. The Fourier-KAN model, in particular, stands out for its clean loss curves and early strength in the classification metrics, despite a slight divergence in recall towards the later epochs.

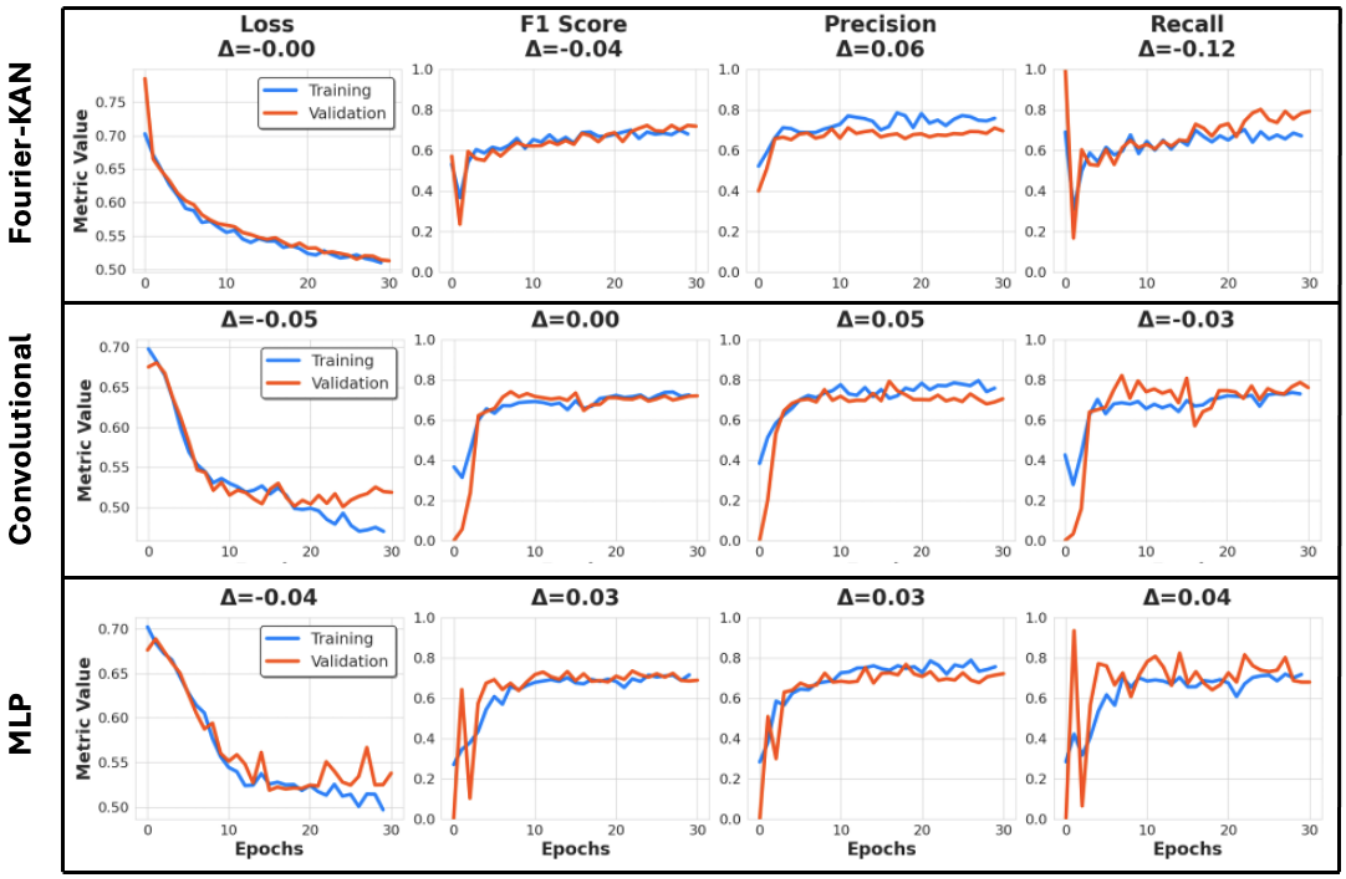

Figure 6: Comparison of Training and Validation Performance Across Different Embedding Variants on Augmented Data.

The Convolutional and MLP models both show a bit more variability, particularly in the loss curves, but still achieve strong convergence in the classification metrics, indicating that they can effectively leverage the augmented data.

Overall, while the Fourier-KAN model maintains its edge in terms of clean and consistent performance, the Convolutional and MLP models also demonstrate solid generalization capabilities, particularly in their ability to adapt to the augmented data and maintain robust performance across multiple evaluation metrics.

5.4 Comparative Analysis

This section provides a side-by-side comparison of the three vision transformer variants based on their performance and explainability. Here, the performance on the test set will be examined for all variants both trained on the original data as well as on the augmented counterpart. The attention maps of the models trained on the original data are thoroughly interpreted and compared to the attention maps of the models trained on the augmented variant, which is to be found in the appendix.

5.4.1 Performance Analysis

In this section, we analyze and compare the performance of the three vision transformer variants—Fourier-KAN, Convolutional, and MLP—on the original and augmented datasets. The analysis is based on four key metrics: Accuracy, F1 Score, Precision, and Recall. Figure 7 presents the side-by-side comparison of the models' performance on the original (right) and augmented (left) datasets.

KAN-Fourier Embedding

On the original dataset, the KAN-Fourier model exhibits strong performance across all metrics, with an Accuracy of 0.754, an F1 score of 0.729, Precision of 0.728, and Recall of 0.745. These metrics reflect the model's robust ability to capture and generalize from the original data, with particularly high recall, indicating effective identification of positive instances.

In contrast, when trained on the augmented dataset, the KAN-Fourier model shows a slight decrease in performance across most metrics. The Accuracy drops to 0.724, F1 score to 0.693, Precision to 0.738, and Recall to 0.677. The decrease in F1 score and Recall suggests that the augmented data might have introduced some noise or complexity that challenged the model's ability to generalize as effectively as it did with the original data. Nonetheless, the model maintains a relatively high Precision, indicating that it still manages to accurately identify positive cases, even if it struggles with Recall.

Convolutional Embedding The Convolutional model performs well on the original dataset, with an Accuracy of 0.765, an F1 score of 0.716, Precision of 0.757, and Recall of 0.694. These results show

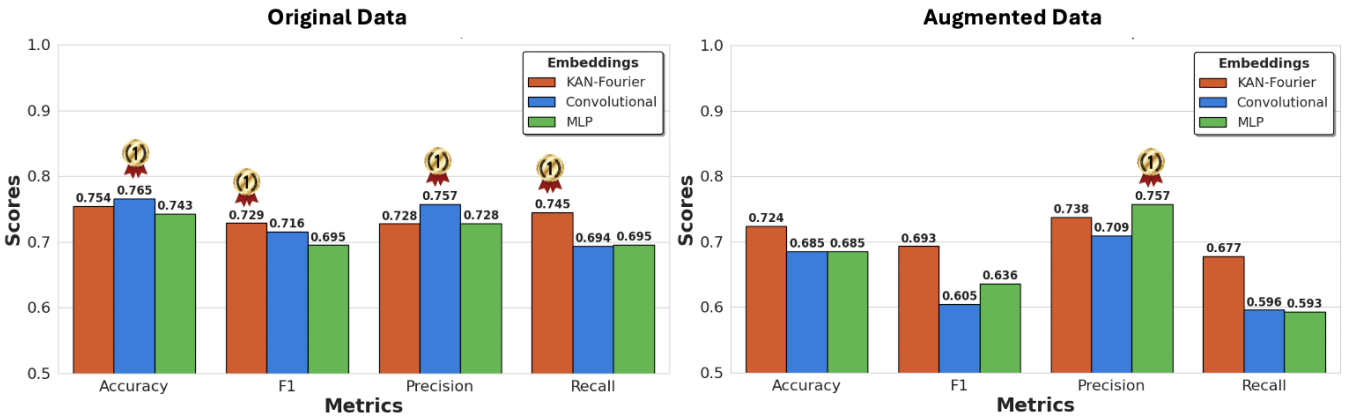

Figure 7: Comparison of Test Set Performance for all Embedding Variants on Original and Augmented Data.

that the model is particularly strong in Precision, suggesting that it is effective at minimizing false positives.

However, the performance of the Convolutional model declines noticeably on the augmented dataset. Accuracy decreases to 0.685, F1 score drops significantly to 0.605, Precision falls to 0.709, and Recall further reduces to 0.596. The substantial drop in F1 score and Recall indicates that the model struggles more with the augmented data, likely due to the increased variability or noise introduced by the data augmentation. This performance gap suggests that the Convolutional model is more sensitive to changes in the data distribution compared to the other models.

MLP Embedding The MLP model demonstrates the weakest performance among the three models on the original dataset, with an Accuracy of 0.743, an F1 score of 0.695, Precision of 0.728, and Recall of 0.695. While these metrics are slightly lower than those of the other models, the MLP model still shows a reasonable balance between Precision and Recall.

When trained on the augmented dataset, the MLP model's performance remains relatively stable compared to the original dataset. Accuracy is 0.685, F1 score is 0.636, Precision improves to 0.757, and Recall drops slightly to 0.593. Although there is a drop in F1 score and Recall, the MLP model's Precision actually improves, suggesting that it might be better at handling the augmented data in terms of identifying true positives, even though it may miss some cases (lower Recall).

Comparison and Conclusion Comparing the performance of the models on the original versus augmented datasets reveals some key insights. Overall, all models experience a decline in performance when trained on the augmented data, particularly in terms of F1 score and Recall. This suggests that while data augmentation can increase the variability and robustness of training data, it also introduces challenges that can impact a model's ability to generalize effectively.

Among the three models, the KAN-Fourier embedding maintains the highest overall performance on both datasets, with the least decline in metrics like Precision. The Convolutional model, while strong on the original dataset, shows the most significant performance drop when applied to the augmented data, indicating a potential sensitivity to the changes introduced by augmentation. The MLP model, despite being the weakest performer on the original dataset, shows a relatively stable performance across both datasets, with an improved Precision on the augmented data.

In conclusion, Figure 8 adds further transparency to the test set performance and can be used to shift decision boundaries as well as serve as a model debugging tool for further fine-tuning and optimization.

5.4.2 Explainability through Attention Maps

In this section, we analyze the attention maps generated by each model variant—Fourier-KAN, Convolutional, and MLP—when applied to the coral images. The attention maps, overlaid on the original images, provide insights into how each model focuses on different aspects of the images to make its predictions. The first row in Figure 9 shows the original images, while the subsequent rows display the averaged attention maps for the Fourier-KAN, Convolutional, and MLP models, respectively. The color scheme used in the attention maps ranges from blue (least attention) to red (most attention).

Fourier-KAN:.

- **Image #1:** The Fourier-KAN model focuses on the fish swimming around the coral reef, with less attention on the healthy corals themselves. Since healthy reefs often feature an abundance of fish, this focus is logical. The model correctly classifies the image with a probability of 0.80.
- **Image #2:** Here, the model directs its attention more towards the reef itself rather than the water, likely due to the absence of fish. It successfully identifies the healthy coral with a probability of 0.70.
- **Image #3:** The model correctly highlights the bleached portions of the coral, confidently classifying it as bleached with a probability of 0.21, despite the presence of a fish.
- **Image #4:** Although the majority of the coral appears healthy, the Fourier-KAN model accurately identifies the tips that are beginning to bleach, highlighting these areas with strong red patches and correctly classifying the coral as bleached with a probability of 0.22.

Convolutional:

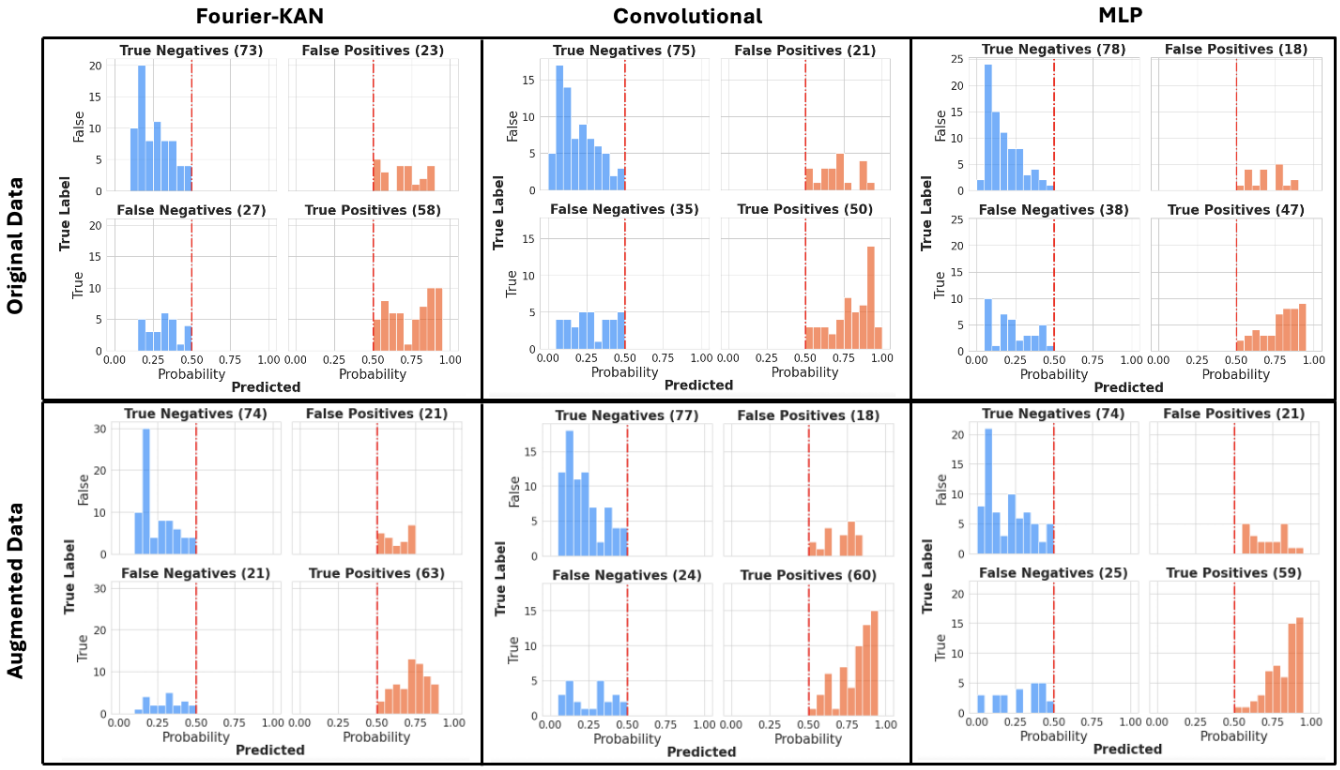

Figure 8: Confusion Matrices displaying Decision Boundaries Predicted and Probability Distributions on Original and Augmented Data (Test Set) across the Embedding Variants.

- **Image #1:** Unlike the Fourier-KAN model, the Convolutional model focuses primarily on the healthy corals, paying little attention to the fish. This leads to a correct classification with a probability of 0.88.
- **Image #2:** The model emphasizes the core of the reef with stronger and more concentrated red patches, resulting in a correct classification as healthy with a probability of 0.78.
- **Image #3:** It successfully highlights the bleached portions of the coral and ignores the fish, leading to a confident classification of bleached with a probability of 0.09.
- **Image #4:** Although it correctly identifies the bleached coral tips, the model also highlights some of the healthy areas, causing an incorrect classification of the coral as healthy with a probability of 0.52.

MLP:.

- **Image #1:** The MLP model focuses on both the reef and the fish, correctly classifying the coral as healthy with a probability of 0.84.
- **Image #2:** The attention patches are somewhat scattered, but the model still manages to classify the coral as healthy with a probability of 0.82.
- **Image #3:** Despite a dispersed attention map, the model confidently classifies the coral as bleached with a probability of 0.21.
- **Image #4:** The model struggles to capture the bleached portions, focusing instead on the healthy areas, leading to an incorrect classification as healthy with a probability of 0.56, similar to the Convolutional model.

Conclusion: Figure 9 provides additional transparency into the decision-making process of each model variant. These attention maps can be used to adjust decision boundaries and serve as a valuable tool for model debugging and further fine-tuning to improve interpretability and performance. Last but not least, the attention maps of the models trained on the augmented data, follow similar patterns, except that those attention maps are more pronounced theough the itroduction of the augmentation techniques as shown in Figure 10, to be found in the appendix.

6 DISCUSSION ON LIMITATIONS

- Openai paper scaling laws: - https://www.youtube.com/watch?v=5eqRuVp65eY
- https://arxiv.org/pdf/2010.14701

7 CONCLUSION AND FUTURE WORK

In this study, we explored the application of Vision Transformers (ViTs) with diverse embedding strategies for the task of coral classification, focusing on limited data scenarios. Our results demonstrated that Kolmogorov-Arnold Fourier embeddings, convolutional embeddings, and linear embeddings can effectively enhance the performance of ViTs in this context. The Fourier-based approach, while

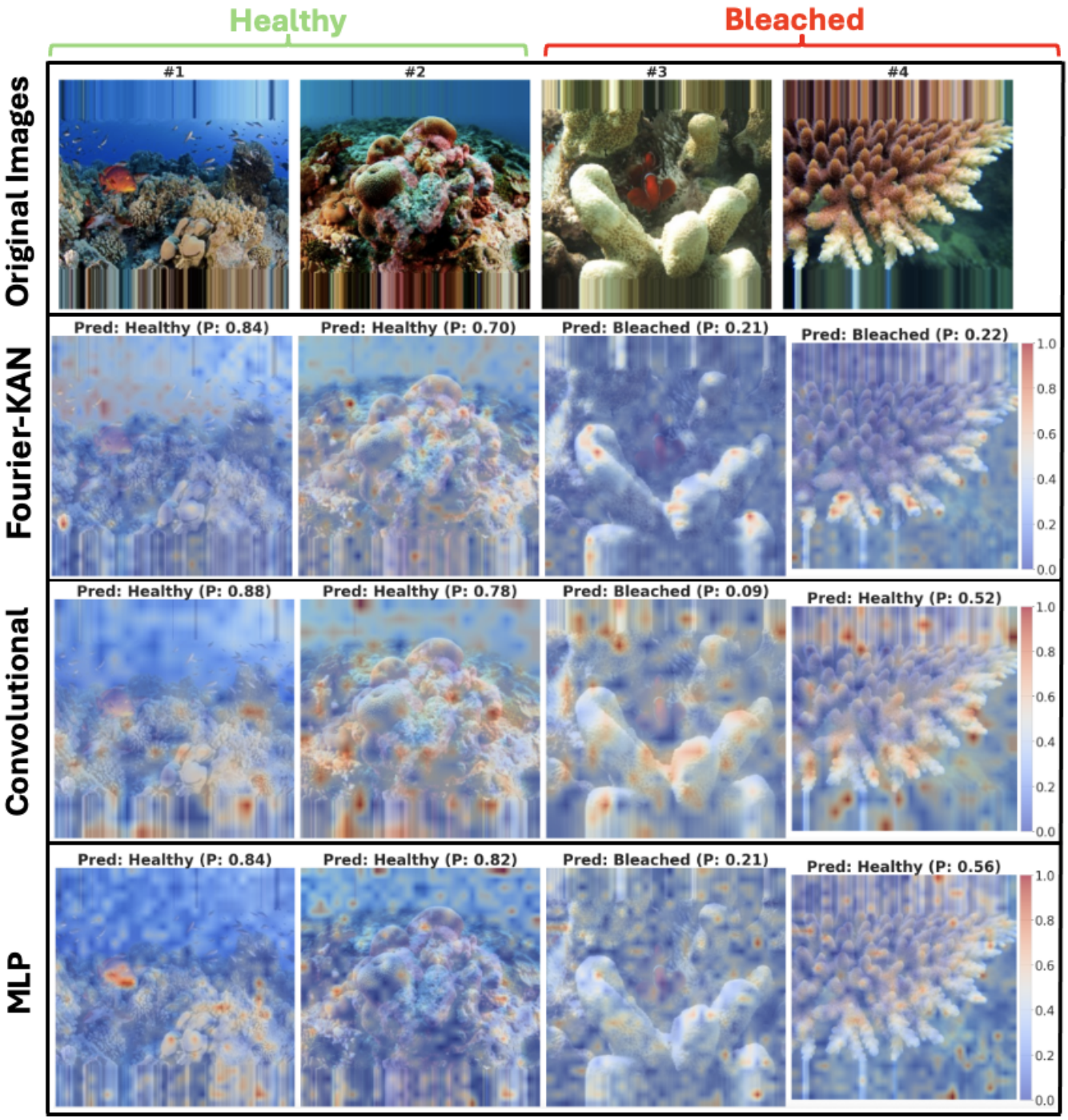

Figure 9: Comparison of Attention Maps Across the Embedding Variants on Original Data.

computationally intensive, showed promising results, especially in capturing complex patterns within coral images. The Convolutional and MLP embeddings also provided strong baseline performances, with unique strengths in different aspects of classification.

However, our experiments were constrained by limited GPU resources, which restricted the extent of hyperparameter exploration and model fine-tuning. Given the computational demands of Kolmogorov-Arnold networks, future work should involve leveraging more powerful GPUs to thoroughly explore the parameter

space, potentially leading to improved model performance and a deeper understanding of the embeddings' impact.

Additionally, we plan to investigate the application of the SWIN Transformer, which could offer further improvements in handling hierarchical features and local-global attention in images, potentially outperforming standard ViTs in coral classification tasks.

Furthermore, expanding the dataset with more diverse and representative images, as well as refining data augmentation techniques, will be crucial for enhancing the robustness and generalization of the models. By simulating a broader range of environmental conditions and coral states, more sophisticated models can be developed, potentially leading to better accuracy in real-world applications.

Another avenue for future research includes integrating multimodal data, such as combining image data with environmental metadata (e.g., water temperature, pH levels), to improve the context-awareness of the models. This integration could enable more accurate predictions of coral health, considering both visual and environmental factors.

Finally, continuing to improve the interpretability of these models, possibly through advanced attention mechanisms or explainable AI techniques, will be essential for gaining trust in the models' predictions, particularly in critical environmental monitoring tasks.

Overall, while this study provides a solid foundation, there are numerous opportunities for further enhancement and exploration to better support coral conservation efforts through advanced deep learning techniques.

A ADDITIONAL FIGURES

REFERENCES

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

[2] A. E. Douglas. 2003. Coral bleaching––how and why? *Marine Pollution Bulletin* 46, 4 (2003), 385–392.

[3] Ove Hoegh-Guldberg, Peter J. Mumby, A.J. Hooten, R.S. Steneck, Patrick Greenfield, Eduardo Gomez, C.D. Harvell, P.F. Sale, A.J. Edwards, Ken Caldeira, et al. 2007. Coral reefs under rapid climate change and ocean acidification. *Science* 318, 5857 (2007), 1737–1742.

[4] Terry P. Hughes, James T. Kerry, Mariana Álvarez-Noriega, Jorge G. Álvarez-Romero, Kristen D. Anderson, Andrew H. Baird, Russell C. Babcock, Maria Beger, David R. Bellwood, Ray Berkelmans, et al. 2017. Global warming and recurrent mass bleaching of corals. *Nature* 543, 7645 (2017), 373–377.

[5] Nain Jamil, MuhibUr Rahman, and Amir Haider. 2021. A Bag of Features Approach for Coral Classification and Localization. *Big Data and Cognitive Computing* 5, 4 (2021), 53.

[6] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. 2024. KAN: Kolmogorov-Arnold Networks. *arXiv preprint arXiv:2404.19756* (2024).

[7] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, G. Kendrick, and R. B. Fisher. 2016. Coral classification with hybrid feature representations. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 519–523.

[8] mintisan. 2024. Awesome KAN (Kolmogorov-Arnold Network). https://github.com/mintisan/awesome-kan

[9] Gist Noesis and unrealwill. 2023. FourierKAN: PyTorch Layer for Fourier Komogorov-Arnold Networks. https://github.com/GistNoesis/FourierKAN GitHub repository, accessed on YYYY-MM-DD.

[10] A. S. M. Shihavuddin, Nuno Gracias, Rafael Garcia, Arthur C. R. Gleason, and Brooke Gintert. 2013. Image-based coral reef classification and thematic mapping. *Remote Sensing* 5, 4 (2013), 1809–1841.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[12] Vencerlanz09. 2023. Healthy and Bleached Corals Image Classification. https://www.kaggle.com/datasets/vencerlanz09/healthy-and-bleached-corals-image-classification Accessed: YYYY-MM-DD.

[13] Jordan M. West and Rodney V. Salm. 2003. Resistance and resilience to coral bleaching: implications for coral reef conservation and management. *Conservation Biology* 17, 4 (2003), 956–967.

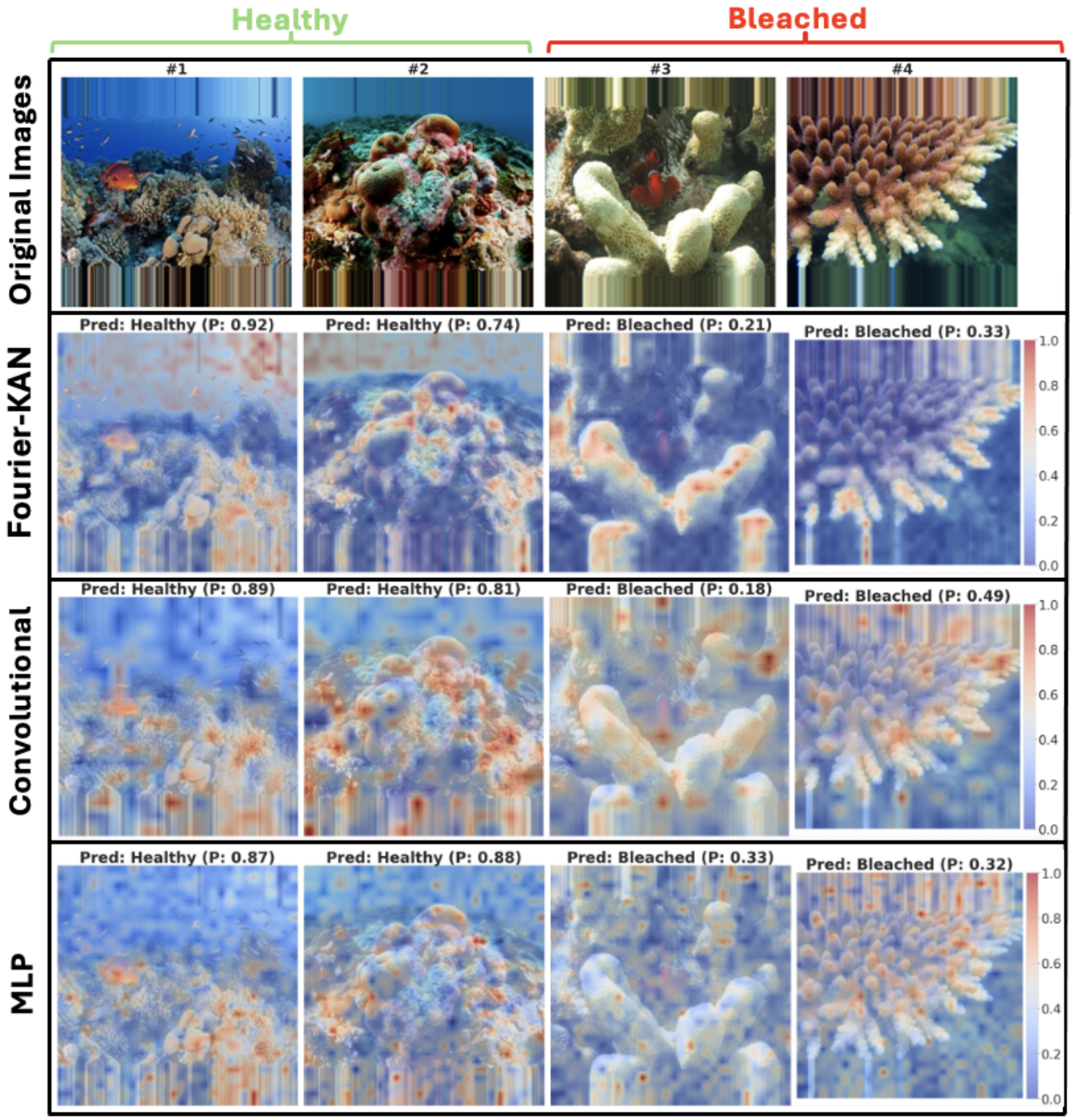

Figure 10: Comparison of Attention Maps Across the Embedding Variants on Augmented Data.