Estimating Water Quality from Satellite Imagery Using the SustainBench Dataset

Diego Zancaneli
Stanford University
diegozan@stanford.edu

Ernesto Sung Woo Nam Song
Stanford University
ernesto.nam@stanford.edu

Rikhil Paresh Vagadia
Stanford University
rvagadia@stanford.edu

Abstract

In this paper, we explore the efficacy of deep learning models in predicting water quality indices from satellite imagery within the SustainBench framework. Specifically, we examine and compare the performance of various Convolutional Neural Network (CNN) architectures, including VGG and ResNet, alongside the Vision Transformer (ViT) model. Our dataset, derived from the Demographic and Health Surveys (DHS), includes multi-channel satellite images and survey-based water quality indices across 49 countries from 1996 to 2019. We preprocess this dataset through rigorous data cleaning and normalization techniques, and train our models on a subset of 26k data points due to computational constraints. Our findings reveal that the ViT model achieves an r^2 of 0.5 in predicting the clean water index, outperforming traditional CNN architectures. This performance is notable given the smaller dataset size used compared to the current leaderboard of the SustainBench dataset (r^2 of 0.4), underscoring the architectural advantages of ViT in effectively leveraging limited data.

1. Introduction

In 2015, the United Nations (UN) introduced 17 Sustainable Development Goals (SDGs) to be achieved by 2030, aimed at fostering prosperity and protecting the planet [1]. Tracking progress towards these goals has relied on costly, infrequent, and mostly inefficient data collection methods, such as civil registrations, surveys, and censuses. This has led to significant data gaps in many countries, hindering effective SDG monitoring [2].

Artificial intelligence has shown promise in addressing these gaps by integrating sparse ground data with abundant, frequently updated sources like satellite imagery, social media posts, and mobile phone activity. To support this effort, SustainBench has been created as a comprehensive set of datasets and benchmarks designed for monitoring SDGs with a focus on computer vision techniques. SustainBench aims to lower the barriers to entry by providing high-quality domain-specific datasets, standardize evaluation benchmarks, and encourage the development of new methods, solving key bottlenecks of computer vision training pipelines. The dataset includes 15 benchmark tasks across seven SDGs, offering crucial resources for the Machine Learning (ML) community to address real-world challenges [3].

We are particularly focused on Clean Water and Sanitation (SDG 6), recognizing universal access to safe drinking water and sanitation facilities as an essential human right. The Sustainable Development Goals underscore the importance of such access, highlighting its role in preventing disease and enhancing human wellbeing. Despite these goals, in 2020, 2 billion people globally lacked safe drinking water, and 2.3 billion did not have access to basic handwashing facilities with soap and water [4]. The link between access to clean water and sanitation and public health outcomes is evident, as improved facilities significantly reduce child mortality rates [5].

2. Related Work

There has been extensive research on predicting water quality levels, initially using structured data. For instance, a paper supported by the Bill & Melinda Gates Foundation employs a Bayesian geostatistical model, integrating geographical coordinates, to generate high-resolution estimates of access to drinking water and sanitation facilities across 88 low- and middle-income countries from 2000 to 2017. By analyzing data from over 600 sources, the study provides detailed subnational insights into geographical inequalities in access, identifying areas needing targeted interventions and successful regions that could serve as models [6].

While still an emerging area, deep learning models are being increasingly utilized to forecast socio-economic indicators using satellite images, which are powerful due to their wide availability and the ability of unsupervised learning to leverage numerous features that would be difficult to collect otherwise [7]. For instance, using publicly avail-

able multispectral satellite imagery, Convolutional Neural Networks (CNNs) were trained to predict survey-based estimates of asset wealth across approximately 20,000 African villages. These models explained 70% of the variation in ground-measured village wealth and up to 50% of the variation in district-aggregated changes in wealth over time, demonstrating the potential of satellite-based estimates to enhance research and policy applications [8].

Other examples include a machine learning approach to accurately identify brick kilns, a major polluting informal industry in Bangladesh, using high-resolution satellite imagery. The method achieves 94% accuracy and 89% precision, providing a low-cost and replicable solution for regulatory agencies to monitor environmental compliance and address pollution sources [9]. Moreover, CNNs were used to predict key livelihood indicators, such as poverty, population, and women's body mass index, from community-generated street-level imagery. The approach demonstrates high classification accuracy and strong r^2 scores for regression in both India and Kenya, highlighting its potential as a scalable, cost-effective alternative to traditional surveying methods [10].

In addition, we are particularly interested in analyzing the results of Vision Transformer (ViT) models as these have not been extensively used in economic development tasks similar to ours. However, one notable example used in a relatively similar field is the paper "Pixel Perfect: Using Vision Transformers to Improve Road Quality Predictions from Medium Resolution and Heterogeneous Satellite Imagery," which demonstrates the efficacy of ViT in accurately predicting road quality from satellite imagery. The authors demonstrate that ViTs outperform traditional CNNs in predicting road quality from medium-resolution satellite imagery. ViTs achieved high AUROC scores of 0.934 for binary classification and 0.685 for five-class classification, making them suitable for infrastructure monitoring in data-scarce regions [11].

3. Dataset and Features

Our goal is to enhance the accuracy of clean water index predictions within the SustainBench's framework by leveraging multi-modal inputs and incorporating the recent advancements in computer vision models we have discussed in class. To do this we will leverage the SustainBench's dataset that pertains to SDG 6 that has been compiled using the Demographic and Health Surveys (DHS), focusing on water quality and sanitation indices across 49 countries [12] from 1996 to 2019 from 56 different countries. This dataset includes labels for 87.9k water quality indices, derived from surveys of 2.1M households. Satellite imagery and DHS data are integrated by using geographically centered satellite images to predict cluster-level water quality indices [3]. SustainBench provides baseline models for each task on a public leaderboard, with the best performance for the water index SDG being a kNN model dating back to 2021, achieving an r^2 of 0.4 [1].

The SustainBench dataset consists of satellite images with dimensions of 255x255 pixels and 8 channels. The first 7 channels represent surface reflectance values from the Landsat 5/7/8 satellites, ordered as follows: blue, green, red, shortwave infrared 1, shortwave infrared 2, thermal, and near infrared. The 8th channel is the nightlights band, sourced from either the DMSP or VIIRS satellite. This detailed multi-channel satellite imagery provides a comprehensive view of the environment, leveraging diverse spectral information to enhance the analysis and prediction tasks. Our labels come from DHS surveys, which can be connected to our satellite images using the DHS-ID. Specifically, for each DHS-ID, we have the DHS year, DHS cluster, latitude and longitude of the cluster, asset wealth index, women's BMI, child mortality rate, women educational attainment, water quality index, and sanitation index. For the purposes of this study, we will restrict our focus to predicting the water quality index for each image.

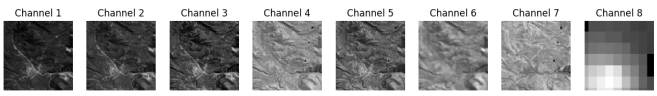

Figure 1. SustainBench Dataset Channels

To prepare our satellite imagery data for model training, we undertook significant preprocessing steps. Initially, we loaded the relevant tar.gz files from a public Google Drive, retaining only the necessary columns needed for our regression task. We further prepared the dataset by extracting additional columns for survey identifiers and paths to the satellite image files. To ensure data integrity, we focused on the data entries with non-missing water_index values and verified that all paths corresponded to existing .npz files in the dataset directory.

We then defined a custom PyTorch Dataset class to handle the loading and preprocessing of the satellite images. Each image, comprising 8 channels, was normalized to the range [0, 1] to better integrate into our various models. If an image had no range in pixel values, it was replaced with zeros to avoid data inconsistencies. NaN values within the images were also replaced with zeros. These preprocessing steps were crucial to ensure the data was clean, consistent, and ready for model training, significantly contributing to the reliability and accuracy of our analysis.

Our group made the decision to use a subset of the whole dataset, consisting of 26,017 data points split into 80% for training, 10% for validation, and 10% for testing, to manage

[1] https://github.com/sustainlab-group/sustainbench/blob/main/baseline_models/dhs/knn_baseline.ipynb

computational and storage constraints effectively. The entire dataset, which comprises 87,938 samples, was too large for our available resources. By using a smaller subset, we were able to optimize our processing capabilities and ensure that the data could be handled efficiently without compromising the integrity and representativeness of our analysis. We are confident that this subset provides sufficient data to train and evaluate our models effectively, maintaining a balance between computational feasibility and robust model performance.

4. Methods and Models

4.1. Baselines

As our baseline models we utilized naive, simple models, namely a k-Nearest-Neighbors (kNN) model and a basic CNN.

The baseline for the clean water index task is a kNN model that inputs the average pixel value for the the nightlights band. The reason we choose the kNN model is because this model was the model that was used in the SustainBench paper for the clean water index metric. Thus, our model was heavily inspired by the specific kNN model built by the SustainBench team who have made their algorithm public on their Github page. Although we use this baseline, we want to point out the model's weakness as it relies solely on a single average pixel value from just one band (the nighlight band) and this fails to holistically represent an 8x255x255 px satellite image. Furthermore, while simple and interpretable, the kNN method often struggles with high-dimensional data and lacks the ability to capture complex patterns in the data effectively.

To address these limitations, we use CNNs due to their superior performance in handling image data and their ability to automatically learn spatial hierarchies of features from input images. Our baseline CNN model consists of two convolutional layers, a max-pooling layer, followed by a fully connected layer (fc1) that maps the flattened output to the desired number of output classes for our regression task. This straightforward architecture is designed for simplicity and ease of understanding, providing a solid foundation for more complex models.

4.2. VGG

To enhance our model's performance, we explore the VGG architecture, which emphasizes depth and simplicity by using small 3×3 convolutional filters stacked on top of each other in increasing depth. This approach allows for a deeper network while keeping the number of parameters manageable. The VGG network, specifically VGG-16 and VGG-19, consists of 16 and 19 weight layers respectively, where each layer is followed by a ReLU activation function and max-pooling layers. The depth and simplicity of VGG networks facilitate efficient learning of complex features, leading to improved performance on image recognition tasks [13].

4.3. ResNet

As a second iteration, we explore the ResNet architecture, which utilizes residual connections to enable the training of deeper networks. The core idea is the use of residual blocks, where the output of a layer is added to the input of that layer after some transformations (mathematically, $\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}$). This architecture allows the network to learn identity mappings more easily, which helps in training deeper networks by mitigating the vanishing gradient issue. ResNets come in various depths, such as ResNet-18, ResNet-34, ResNet-50, and ResNet-101, each with an increasing number of layers and complexity [14].

4.4. Vision Transformer (ViT)

In addition to the CNN models we used, another model we heavily rely on is the ViT model which was first introduced by Dosovitskiy et al. in their paper "An image is worth 16x16 words: Transformers for image recognition at scale" that is an adaption of the Transformer architecture commonly used in the field of Natural Language Processing [15] [16].

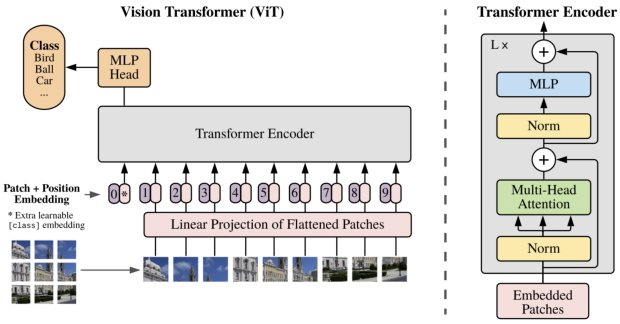

Figure 2. Vision Transformer Model

The standard Transformer is designed for text data and accepts a one-dimensional input of word token embeddings. To adapt a Transformer to image data, the ViT processes a three-dimensional image $x \in \mathbb{R}^{H \times W \times C}$, where H and W are the height and width, and C is the number of channels. ViT extracts N non-overlapping image patches, turning it into a sequence of patches $x_p \in \mathbb{R}^{N \times P \times P \times C}$, where (H, W) is the height and width of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = \frac{HW}{P^2}$ is the resulting number of patches, which also serves as the effective input sequence length for the Transformer. These patches are then linearly projected into an $N \times D$ space to obtain a feature extraction z_0, which

includes positional encodings. A ViT model is comprised of 3 main components, a linear layer for patch embedding, a stack of transformer blocks with multi-head self-attention and feed-forward layers for feature encoding, and a linear layer for classification score predictions which can be seen in the figure below.

ViT models offer several advantages over traditional CNNs as their self-attention mechanisms capture long-range dependencies and global context within images (vs. CNNs, which rely on local receptive fields). This results in less image-specific inductive bias, as CNNs are dependent on the neighborhood structure within the two-dimensional $H \times W$ space. In contrast, the fully connected nature of ViT's multi-headed self-attention layers allows for a global understanding of the image. The ViT model we used is based on the google/vit-base-patch16-224 configuration, which has been pretrained on the ImageNet-1k dataset. This dataset comprises 1,000 classes and over 1.2 million images, providing a comprehensive base for transfer learning, having been tweaked as per our specific modifications and extensions.

5. Experiments

5.1. Experimental Details

We trained our models using the Adam optimizer with a cosine learning rate scheduler, starting at a learning rate of 1e-2 and an epsilon of 1e-8. These settings were finalized after extensive hyperparameter tuning. The loss function for all models is Mean Squared Error (MSE), which is suitable for regression tasks. Each model was trained for 10 epochs. Google's Colab was utilized for most code files, with training conducted primarily on Colab's T4 and L4 GPUs.

5.2. Evaluation Method

To evaluate the performance of our models, we employ three key metrics: the coefficient of determination (R^2), MSE, and Mean Absolute Percentage Error (MAPE).

The R^2 metric measures the proportion of the variance in the dependent variable that is predictable from the independent variables:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where y_i are the true values, $\hat{y}_i$ are the predicted values, and $\bar{y}$ is the mean of the true values. An R^2 value close to 1 indicates that the model explains a large portion of the variance in the data, making it a useful metric for assessing model performance in regression tasks like predicting the clean water index.

MSE quantifies the average squared difference between the predicted values and the actual values:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

MSE is particularly useful because it penalizes larger errors more significantly, making it sensitive to outliers. Lower MSE values indicate better model performance. MSE provides a clear understanding of how well the model's predictions align with the actual values on average.

MAPE measures the average absolute percentage difference between the predicted values and the actual values:

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

By using R^2, MSE, and MAPE, we obtain a comprehensive evaluation of our model's performance. These metrics together provide insights into the variance explained by the model, the average error magnitude, and the overall prediction accuracy, ensuring a robust assessment of the model's effectiveness in predicting the clean water index.

Note that we initially relied on MSE and MAPE for our first experiments to maintain consistency and comparability across different models. However, we incorporated R^2 in our ViT experimentation because our primary goal was to highlight the advantages of the ViT model. By focusing on R^2 for ViT, we aimed to showcase its superior performance and alignment with the baseline metrics, demonstrating its value over traditional models like CNNs, which have been explored in the literature before. We decided to include R^2 only for ViT, which was the model we intended to submit to the SustainBench leaderboard especially since the R^2 values for VGGNet and ResNet were lower than those for ViT nonetheless.

.

5.3. Baseline

Table 1. Comparison of Baseline Models

Model	R^2	**MSE**	**MAPE**
kNN	0.281	0.756	.238
Basic CNN Model	0.1574	1.0766	.3494

The k-Nearest-Neighbors (kNN) model, inspired by the SustainBench approach, showed an R^2 of 0.281, MSE of 0.756, and MAPE of 0.238. While simple and interpretable, it struggles with high-dimensional data, relying solely on the night-lights band, which fails to capture the full complexity of the 8x255x255 px satellite images. The basic CNN model, designed with two convolutional layers and a max-pooling layer, yielded an R^2 of 0.1574, MSE of 1.0766, and MAPE of 0.3494. Despite its potential to learn

spatial hierarchies, the CNN's performance indicates that a more sophisticated architecture is needed to improve prediction accuracy for the clean water index.

5.4. VGG

VGG was originallly designed for image classifications tasks and is renowned for its simplicity and depth, which contribute to its high performance. We adapted VGG's architecture to enhance its capability for predicting water indices from satellite imagery, a regression task. Specifically, we utilized the pre-trained VGG16 model, a variant within the VGG family that typically includes layers configured for classifying images into 1,000 categories. We removed the final fully-connected classifier layer and replaced it with a regression layer tailored to output continuous values reflective of water indices. This modification allowed the model to transition from classification to regression, leveraging the rich feature extraction capabilities of VGG while aligning with our specific task.

The VGG architecture comes in various configurations, differing primarily in depth—specifically, the number of convolutional layers. The most commonly discussed variants are VGG-16 and VGG-19. The difference between them lies in the addition of three extra convolutional layers in VGG-19, which can potentially offer deeper and more nuanced feature detection at the cost of increased computational overhead.

Table 2. Comparison of VGG Models

Model	MSE	MAPE
VGG13	0.54	0.3418
VGG16	0.317	0.106
VGG19	0.2252	0.0793

The choice between VGGNet-16 and VGGNet-19 often hinges on a trade-off between performance and computational efficiency. We chose to conduct most of our experiments with VGGNet-16 due to its relatively lower complexity and faster training times, which proved sufficient for our needs in extracting relevant features from satellite images for water index prediction. This choice was further validated by preliminary tests that indicated VGGNet-16 provided a favorable balance between accuracy and performance.

We experimented with three different pooling methods—average pooling, max pooling, and global average pooling—to enhance the feature extraction capabilities of our VGGNet-16 model for predicting water indices from satellite images. While max pooling is the default setting in VGGNet and focuses on the most prominent features, we also explored average pooling and global average pooling. Average pooling helps in smoothing out image features, useful for capturing overall patterns that are more robust to noise. Global average pooling reduces each feature map to a single value, simplifying the model and improving computational efficiency. By testing these methods, we aimed to determine which pooling technique best balances detailed feature capture with generalization across different images, ensuring our model is both accurate and efficient for practical applications, as seen below.

Table 3. Experimenting with VGG Pooling Layers Modification

Model	MSE	MAPE
Max Pooling	0.317	0.106
Avg Pooling	0.3857	0.161
Global Avg Pooling	0.391	0.138

Inspired by the VGG family models that adjust the depth of their layers, we experimented with complexity adjustments through the development of two distinct experimental architectures to enhance the predictive capability of our models. In the first experiment, called VGGModifiedClassifier, we incorporated additional fully connected layers to the base VGG-16 architecture, aiming to determine how increasing depth affects feature processing. This model starts with the pre-trained VGG-16 feature extractor, followed by a series of dense layers each comprising 4096 neurons, ReLU activation, and dropout for regularization. The flexibility to add more layers allows us to explore different model complexities and their impact on learning detailed patterns in the data.

The second experiment Multiscale Feature Fusion approach, where we made a class called VGGMultiScale to extract and combine features from two different layers of the VGG-16 model. By selecting features from mid-conv3 and conv4 layers, this architecture harnesses information from both mid-level and deeper network stages, thereby capturing a broader range of image details. The extracted features from each scale are flattened and then concatenated to form feature vector that is subsequently processed through linear layer. This layer integrates these multiscale features into a final output, designed to improve the model's ability to discern subtle variations. This method not only leverages the inherent strengths of multiscale feature extraction but also optimizes the model for enhanced predictive accuracy and computational efficiency. The results from these experiments, as summarized in the table below, indicate that while VGGMultiscale showed slightly better efficiency and accuracy versus the baseline VGG16, VGGModifiedClassifier was still outperformed by our baseline. Still, neither surpassed VGG-19. Two possible hypotheses for this outcome could be that the added complexity in VGGNet-19 captures finer details more effectively for satellite imagery analysis, or that our modifications did not

sufficiently align with the intricate patterns and variability present in the data set.

Table 4. Experimenting with Adjusted VGG Layers

Model	MSE	MAPE
VGGModifiedClassifier	0.377	0.122
VGGMultiScale	0.349	0.11

5.5. ResNet

We started with a pre-trained ResNet-18 model from PyTorch's torchvision library, fine-tuned for our specific use case given its proven performance in image classification task.

Just like in the VGG case, our preprocessing pipeline is designed to handle large satellite image datasets stored in .npz files, resizing them to 224x224 pixels to match the input size required by ResNet-18. Additionally, the dataset undergoes principal component analysis (PCA) to reduce dimensionality, retaining three principal components to simplify the data without significant loss of information. This step helps in managing the high-dimensional satellite image data more efficiently. The use of data loaders for batching and shuffling ensures efficient training and evaluation processes. The training loop is structured to leverage GPU acceleration when available, optimizing the model using the Adam optimizer and tracking the performance through training and validation splits.

We experimented with different ResNet architectures, specifically ResNet-18, ResNet-34, ResNet-50, and ResNet-101. ResNet-50 and ResNet-101 use more complex bottleneck blocks consisting of three layers each. Deeper networks like these can learn more complex features, potentially improving performance on challenging tasks, but they also require more computational resources and are more prone to overfitting.

Our experiments showed that ResNet-101 achieved the best results, with an MSE of 0.3697 and a MAPE of 0.1091, outperforming the other ResNet variants. The superior performance of ResNet-101 is due to its ability to capture intricate patterns and features within the satellite imagery. Despite the risk of overfitting, our well-designed preprocessing and training pipeline mitigated this issue, enabling ResNet-101 to effectively leverage its depth for the most accurate predictions.

Table 5. Comparison of ResNet Models

Model	MSE	MAPE
ResNet-18	0.4706	0.1229
ResNet-34	0.3939	0.1091
ResNet-50	0.3808	0.1165
ResNet-101	**0.3697**	**0.1091**

We then fixed ResNet-101 as our choice and experimented with average pooling (versus the default max pooling for the ResNet architecture). The idea was that average pooling could be beneficial for tasks where overall patterns and textures matter, as it averages the features, potentially making the model more robust to noise and providing a more generalized representation of the features. However, similar to the VGG case, average pooling didn't change the results much, so we kept max pooling.

Table 6. Experimenting with Max vs Avg Pooling in ResNets

Model	MSE	MAPE
Max Pooling	0.3697	0.1091
Avg Pooling	0.3674	0.1086

Next, we tried adding dropout layers to my model to improve generalization and stabilize training by preventing overfitting. Dropout randomly deactivates a fraction of neurons during training, encouraging the model to learn more robust features. However, the results were not as intuitive as expected. With a dropout rate of 0.5, the model's performance significantly deteriorated, showing higher MSE and MAPE, suggesting over-regularization. A lower dropout rate of 0.1 also did not improve performance compared to no dropout, indicating that either the network did not benefit from dropout or that optimal regularization might be achieved through different techniques.

Table 7. Experimenting with Dropout for ResNets

Model	MSE	MAPE
No dropout	**0.3697**	**0.1091**
0.5 dropout	1.4709	0.2314
0.1 dropout	0.4247	0.1117

Finally, using the original simpler ResNet-101 architecture, we explored different approaches for handling image channels. We tried simplifying the PCA approach to using just the three most informative channels that we saw are typically used in satellite imagery tasks (the RGB channels). This helped us understand the impact of dimensionality reduction on model performance and led to the optimal selection of input channels.

Table 8. Experimenting with Channels used for ResNets

Model	MSE	MAPE
All 8 channels	0.3697	0.1091
3 RGB channels	**0.2998**	**0.0939**

5.6. ViT

Based on the satellite imagery literature, we implemented key modifications to enhance our original ViT model.

First, we incorporated Patch Merging layers into the standard ViT model, inspired by techniques described in the paper "Vision Transformer for Multispectral Satellite Imagery: Advancing Landcover Classification" [17]. The Patch Merging mechanism progressively reduces the spatial dimensions of the patch embeddings while increasing their feature dimensions, allowing the model to efficiently process high-resolution satellite images. Our custom PatchMerging class merges adjacent patches into larger patches, reducing the spatial resolution by a factor of two in each dimension and increasing the feature dimensions using a linear projection. In the enhanced ViT model, initial patch embeddings are obtained from the standard ViT, followed by two Patch Merging layers applied sequentially. This process reduces the number of patches and enriches the feature representation at each step. After patch merging, global average pooling aggregates the features across all patches, and a fully connected layer produces the regression output. Additionally, we replaced the original classifier head of the ViT with a custom classifier consisting of a single linear layer to output one continuous value for regression purposes. The ViT model layers were frozen, except for the newly added regression head. We also used the pretrained feature extractor of the ViT.

Mixed precision training (fp16) was enabled to speed up the training process, and gradient accumulation was set to 2 steps to handle larger batch sizes effectively. Furthermore, we chose to use global average pooling as well as dropout applied both after patch embeddings and within transformer encoder layers, as these approaches consistently yielded better results and helped reduce overfitting.

In addition, extensive hyperparameter tuning was conducted to identify the optimal settings for learning rates, batch sizes, and optimizer type. By experimenting with the model's depth and width, we adjusted the number of Transformer layers and their size to balance computational efficiency with accuracy. This included results for a model approximately 3.56 times larger in terms of the number of parameters, with 33% wider hidden layers, twice as many layers, and 33% more attention heads compared to the original model. Like in VGG and ResNets, we explored different approaches for handling image channels, running experiments using both (i) all 8 channels and (ii) the RGB channels only. We also experimented with global average pooling, global max pooling, and combining global average and max pooling, as well as applying dropout at various stages of the model architecture, such as after linear layers, attention layers, and pooling layers.

These enhancements, inspired by our thorough literature review and rigorous experimentation, allowed our ViT model to significantly improve its performance in predicting the clean water index. For brevity, the results presented below focus on the most promising configurations.

Table 9. Comparison of Vision Transformer Models

Model	R^2	MSE	MAPE
With PCA	.21	.876	.239
With RGB Channels	.50	.799	.266
With RGB and Patch Merging	.35	.821	.2888
Deeper and Wider Model With RGB	.33	1.01	.3286

In our ViT model for predicting the clean water index from satellite imagery, using only the 3 RGB channels yielded better performance compared to employing PCA. The RGB channels capture high-quality spatial information crucial for distinguishing land cover types and identifying features like water bodies, vegetation, and built-up areas, which are essential for assessing water quality. PCA may create new components that do not align with the specific features needed for our task, potentially diluting important spectral characteristics. These findings align with the overall literature, where RGB data is often sufficient for many tasks due to its high information content and direct visual correspondence (as well as with the results found in our VGG and ResNet experiments). The ViT model using only the RGB channels yielded the best performance, with an R2 value of 0.50, MSE of 0.799, and MAPE of 0.267. In comparison, the ViT model with PCA showed an R2 of 0.21, MSE of 0.876, and MAPE of 0.239.

Increasing the depth and width of the ViT model resulted in an R2 of 0.33, MSE of 1.01, and MAPE of 0.329, possibly due to over-parameterization with the relatively small dataset.

These findings emphasize that using only the RGB channels provides the most effective data representation for predicting the clean water index. While PCA and patch merging introduce beneficial dimensionality reduction and spatial structuring, they do not outperform the straightforward RGB-only approach. Additionally, increasing model complexity with deeper and wider architectures does not yield better results, emphasizing the importance of aligning data preprocessing techniques and model complexity with the specific characteristics of the task.

5.7. Error Analysis

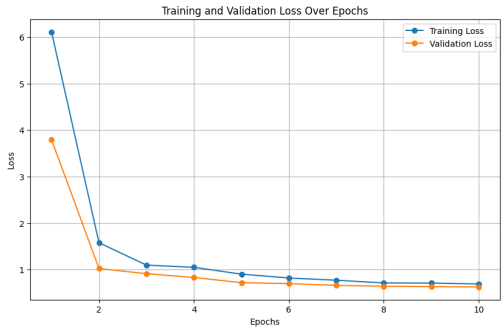

Figure 3. Training and Validation Loss Over Epochs For Best R^2 ViT Model

Figure 3 below shows that both the training and validation losses decrease rapidly in the first few epochs and then stabilize, indicating that the ViT model is effectively learning and generalizing well to the validation data.

6. Conclusion & Future Work

In this study, we investigated the effectiveness of various deep learning models in predicting water quality indices from satellite imagery using the SustainBench dataset. We explored CNNs like VGG and ResNet, along with the ViT model. Our dataset, derived from the DHS, included multichannel satellite images and survey-based water quality indices from 49 countries spanning 1996 to 2019. Through data cleaning, normalization, and preprocessing, we trained our models on a subset of 26k data points due to computational constraints. Our experiments revealed that the ViT model, using only the RGB channels, achieved the best performance with an r^2 of 0.5, MSE of 0.799, and MAPE of 0.267, surpassing the current SustainBench leaderboard's best kNN model with an r^2 of 0.4 despite using less datapoint for training compared to the benchmark.

The ViT model's superior performance can be attributed to its self-attention mechanisms, which capture long-range dependencies and global context within images, offering less inductive bias compared to CNNs. This global understanding allowed ViT to better leverage limited data, making it more effective for our task. In contrast, deeper and wider architectures, such as the enhanced ViT and ResNet-101, did not yield better results, likely due to overparameterization and the relatively small dataset size. Additionally, PCA and patch merging, while beneficial for dimensionality reduction and spatial structuring, did not outperform the straightforward RGB-only approach. These findings highlight the importance of aligning data preprocessing techniques and model complexity with the specific characteristics of the task.

For future work, with more time and specially computational resources, we would explore several avenues. First, increasing the dataset size (i.e., using the full dataset) or utilizing data augmentation techniques could help mitigate overfitting and improve model generalization for the simpler models. Second, experimenting with advanced ensemble methods that combine the strengths of different architectures could yield better performance. Third, exploring more sophisticated data integration methods, such as combining satellite imagery with additional socio-economic data, could enhance predictive accuracy. Lastly, many other SDG-related metrics could be predicted using the same deep learning strategies, so we could replicate the analysis to other contexts to increase robustness and also add more value in the application realm.

7. Contributions & Acknowledgements

All team members actively participated in the brainstorming process to define our research direction. Diego and Ernesto explored an alternative path using LPR camera data from a company in Brazil, which ultimately did not yield promising results, whereas Rikhil focused on the SustainBench path from the beginning. For the implementation, we divided the tasks: Ernesto led the VGG efforts, Diego focused on ResNet, and Rikhil concentrated on ViT. Additionally, Ernesto undertook the exploration and setup of our EC-2 instances from AWS. Everyone contributed to writing the report, ensuring a comprehensive and collaborative effort. Given the complexity of the ViT implementation, Diego and Ernesto helped with the broader sections of the report to ensure a balanced workload.

References

[1] United Nations. Transforming our world: The 2030 agenda for sustainable development, Sep 2015.

[2] Marshall Burke, Anne Driscoll, David B. Lobell, and Stefano Ermon. Using satellite imagery to understand and promote sustainable development. *Science*, 371(6535):eabe8628, 2021.

[3] Christopher Yeh, Chenlin Meng, Sherrie Wang, Anne Driscoll, Erik Rozi, Patrick Liu, Jihyeon Lee, Marshall Burke, David B. Lobell, and Stefano Ermon. Sustainbench: Benchmarks for monitoring the sustainable development goals with machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[4] Jeffrey Sachs, Christian Kroll, Guillaume Lafortune, Grayson Fuller, and Finn Woelm. *Sustainable Development Report 2021*. Cambridge University Press, 2021.

[5] G. Fink, I. Günther, and K. Hill. The effect of water and sanitation on child health: evidence from the demographic and health surveys 1986-2007. *Int J Epidemiol*, 40(5):1196–1204, October 2011. PMID: 21724576.

[6] Local Burden of Disease WaSH Collaborators. Mapping geographical inequalities in access to drinking water and sanitation facilities in low-income and middle-income countries, 2000-17. *Lancet Glob Health*, 8(9):e1162–e1185, September 2020. PMID: 32827479; PMCID: PMC7443708.

[7] Sauren Khosla, Benjamin Wittenbrink, and Caroline Zanze. Predicting maternal and infant health outcomes in western africa using satellite images, 2023.

[8] Christopher Yeh, Anthony Perez, Anne Driscoll, George Azzari, Zhongyi Tang, David Lobell, Stefano Ermon, and Marshall Burke. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature Communications*, 11:2583, 2020.

[9] Jihyeon Lee, Nina R. Brooks, Fahim Tajwar, and Stephen P. Luby. Scalable deep learning to identify brick kilns and aid regulatory capacity. *Proceedings of the National Academy of Sciences*, 118(17):e2018863118, 2021.

[10] Jihyeon Lee, David Grosz, Burak Uzkent, Sherrie Zeng, Marshall Burke, David Lobell, and Stefano Ermon. Predicting livelihood indicators from community-generated street-level imagery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):268–276, 2021.

[11] Aggrey Muhebwa, Gabriel Cadamuro, and Jay Taneja. Pixel perfect: Using vision transformers to improve road quality predictions from medium resolution and heterogeneous satellite imagery. *Association for Computing Machinery*, 1(1), 2023.

[12] ICF. Demographic and health surveys (various). Funded by USAID, 1996-2019. Data collection spanning from 1996 to 2019.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[15] 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. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2021.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, abs/1706.03762, 2017.

[17] Ryan Rad. Vision transformer for multispectral satellite imagery: Advancing landcover classification. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024:8176–8183, 2024.