Benchmarking Methane Emission Detection through Satellite Images

Alex Herron, Dhruv Saxena, Xiangyue Wang
ah5865@nyu.edu, ds6802@nyu.edu, xw1499@nyu.edu
Center for Data Science, New York University
Capstone Mentor: Mark Ho
Group 10, Project 5
Orbio Earth

Abstract

In this capstone project, we constructed an image segmentation model that identifies and masks the extent of methane emissions using satellite images of methane plumes. Using simulated data provided by Orbio Earth, we learned that a statistical threshold method is currently better suited for methane plume masking than a U-Net or a pre-trained ResNet model.

1 Introduction

Man-made methane is a greenhouse gas 84-87 times as potent as CO_2 when considering its impact over a 20-year time frame and the second most abundant greenhouse gas in the atmosphere [5]. As a result, methane is a substantial contributor to global warming, but it is also a key to tackling it. Because methane stays in the atmosphere for much shorter periods than CO_2 (around 12 years, compared to centuries for CO_2), any reduction in methane emission today can have a relatively quick effect in reducing global warming [5]. In fact, cutting methane emissions in half by 2030 alone can avoid 0.5°C of global warming by the end of the century [3].

In order to significantly reduce methane emissions today, precise and frequent measurements are necessary to keep companies and nations accountable for their emissions and benchmark their emission reduction efforts. In 2018, the NASA Decadal Survey placed the identification and understanding of methane emissions as one of the top priorities in methane reduction [2]. However, the energy sector consistently under-predicts methane emissions by 50-70% [8].

The challenges associated with methane emission accounting often fall into two categories: precision and scale. Since man-made methane sources such as oil and gas wells are often point-sources, a high degree of precision is required to identify those sources and quantify their emissions. Researchers have utilized surface monitoring networks consisting of sensors to gain precise on-site methane measurements, but such bottom-up approaches suffer from the limitation of scalability [1]. Oil and gas facilities are numerous around the world, and a significant portion of abandoned facilities continues to emit methane. In order to measure methane at the asset, company, or state level, the measurement method must be widely applicable.

Remote satellite sensing is one way to quantify methane emissions at scale [10]. Airborne imaging using absorption spectrometry enables the detection of methane emissions by utilizing bandwidths of short-wave infrared light that are particularly sensitive to methane. The resulting images make methane gas, and their corresponding plumes, visible to both human eyes and computer vision algorithms. Coupled with the increasing number of methane-focusing satellites launching into space, remote satellite sensing opens up a world of potential for scalable and accurate quantification of methane emissions [12]. To quantify methane emission from satellite images, one must create

accurate masks of methane plumes. Our work focuses on this task using both a statistical method and Convolution Neural Networks (CNNs). We sought to improve the accuracy of global methane emission quantification by constructing computer vision models that quickly and accurately mask methane plumes in satellite images. Based on the data available to us, we learned that traditional statistical methods are better suited for methane plume masking than CNN methods.

2 Related Work

We followed the lead of recent work in applying CNNs to quantify methane plume emissions from 2-D high-resolution imagery. Specifically, we were inspired by MethaNet, a CNN model built by Jongaramrungruang, Thorpe, Matheou and Frankenberg to quantify methane point-source emission [9]. The authors of MethaNet used a straightforward model architecture consisting of fifty layers of convolution, two layers of max pooling, and two fully connected layers. Like all other machine learning models, MethaNet requires a large amount of data to train. Due to the limitation of validated, masked methane plume satellite images for training, the authors of MethaNet decided to train their model using a large dataset of simulated methane plumes. Similar to Jongaramrungruang et al., we also used simulated methane plume data, which was provided to us by our partner Orbio Earth.

3 Problem Definition and Algorithm

3.1 Task

Our main task was a classic usage of computer vision: image segmentation. The inputs were simulated satellite images, which consisted of B11 and B12 bands of the light spectrum, and contained snapshots of a potential methane-contributing asset. First, we performed various preprocessing enhancements on each image. Then, after feeding the preprocessed image into our model, we produced a methane signal image of the same size as the input. In our output, the masked pixels were binary values highlighting the presence of methane.

We discovered a number of possible approaches to perform our task. We shortlisted methods that were fundamentally distinct from one another in order to gain a comprehensive perspective on the dataset and its nuances.

First, we noticed the usage of U-Net architectures for image segmentation problems (such as brain tumor detection) and chose to create such a model for this task. One of the advantages of using U-Net was its ability to learn representations in the image quickly. Additionally, we noted the robustness of U-Net models for out-of-sample testing. However, training the U-Net on a large dataset presented its own set of problems in terms of the required computational power and lack of interpretability of its layer-by-later outputs.

This prompted us to try out some state-of-the-art pretrained models for image segmentation. These models were pre-trained on massive datasets, and showed potential to learn the representations faster. However, these models weren't pre-trained on satellite images. Consequently, they were not able to deliver on out-of-sample datasets.

Finally, we constructed a statistical model, rather than a neural network. This model did not require training, and instead relied on the statistics (standard deviation and median) of the values in the input image to create a threshold. Then, a mask was created by converting all values below this threshold to 0s and all values above the threshold to 1s. We noticed that our input images were highly skewed as most of the image area didn't contain methane. This method was able to rectify the interpretability problems we faced with previous neural networks and was scalable and far simpler.

3.2 Algorithm

The first model we implemented was a U-Net model, which we chose for its proven track record with segmentation problems. These models feature a U-shaped model architecture where the input image is gradually compressed into a lower-dimensional representation and then decompressed [4] (see Figure 1). The purpose of compression is to allow to model to focus on both low-level details and high-level abstractions. To utilize both pieces of information during predictions, U-Net also features skip connections between the two sides of 'U', which are direct channels that allow

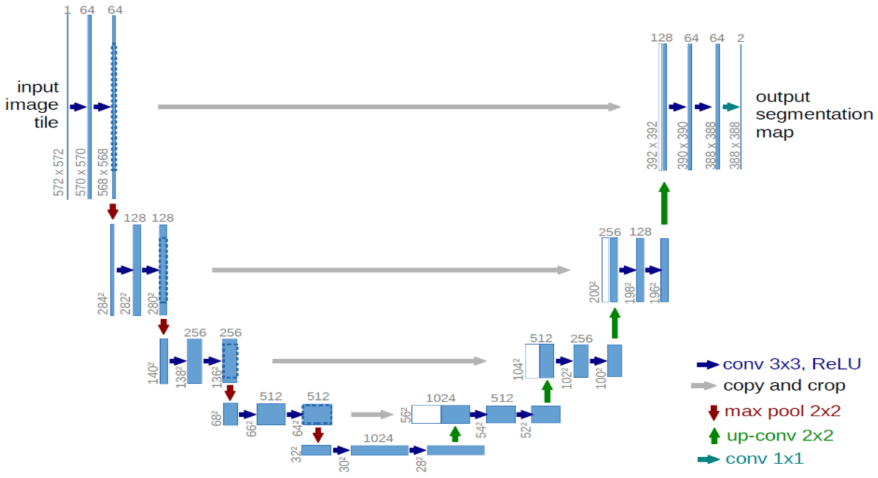

Figure 1: U-Net Architecture [4]

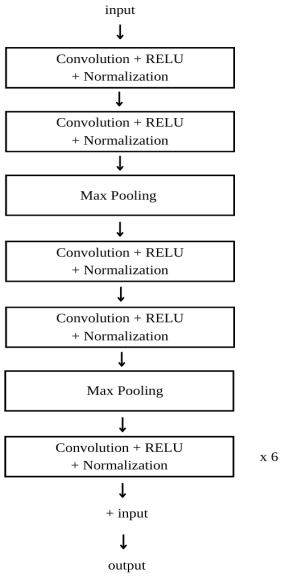

Figure 2: Block model architecture of the Modified U-Net model we constructed.

intermediate inputs to skip certain layers of convolution and be added to the outputs instead. Doing so makes forward and background propagation smoother and allows the model to add more layers. We experimented with both a standard U-Net architecture, as well as a modified U-Net architecture that included additional convolutional layers and skip connections. With the modified U-Net architecture, we expanded each block of convolutional layers in the original U-Net from three layers to ten layers using the architecture shown in Figure 2. After each convolution, there is normalization, which greatly accelerates training, improves regularization, and makes the model less sensitive to initialization. There are also two layers of max pooling and a skip connection added. By expanding each block in the U-Net, we expanded the depth of the entire U-Net to 50 layers.

For both U-Net models, we applied a median split threshold to filter out weak predictions. However, the results of the standard U-Net architecture turned out to be superior. Consequently, we only proceeded with hyperparameter tuning for our standard U-Net model.

Additionally, we experimented with a pre-trained ResNet-101 model [7]. The model consists of 101 layers of convolutions, normalizations, and skip connections.

Lastly, we constructed a simple statistical model using the algorithm detailed in Algorithm 1. For this model, we calculated the median $\tilde{x}$ and standard deviation σ, of the values in each input image. We then picked a scaling factor K, which we later used to hypertune the model. Lastly, we defined a threshold (T) as:

$$T = \tilde{x} + K * \sigma$$

We set the mask values of all methane signals above the threshold as 1s, and all values below as 0s. We initially built this model to use as a baseline, but to our surprise, it performed exceptionally well on the data.

Algorithm 1 Statistical Threshold

Input = Methane Images
Output = Methane Plume Masks
for every image **do**
$\tilde{x}$ = median methane signal
σ = standard deviation of methane signals
K = scaling factor
$T = \tilde{x} + K * \sigma$
    for every pixel **do**
        if *methane signal* $> T$ **then** *mask* $= 1$
        if *methane signal* $< T$ **then** *mask* $= 0$

4 Experimental Evaluation

4.1 Data

For this project, the data consisted of simulated satellite images, simulated ground truth masks, and images consisting of various short-wave infrared bandwidths. The simulated satellite images, or "raw retrievals", were NumPy files consisting of 250 x 250-pixel images. These retrievals did not include noise, and simply included the simulated methane signatures on a blank background. These retrievals served as the inputs for all our models. The goal of this project was to create binary masks from these retrieval images. We trained and evaluated our models using the simulated ground truth masks. Similar to the retrievals, these masks were NumPy files with dimensions of 250 x 250. However, rather than including a spectrum of continuous values, these masks only consisted of 0s and 1s. Each of these binary pixels identified the presence or absence of methane. Additionally, the data included simulated images of short-wave infrared bandwidths (including both B11 and B12 bands), formatted as tiff files of dimension 250 x 250. Both of the bands correspond to short-wave infrared wavelengths, and have a 20-meter resolution. However, the B11 band corresponds to a central wavelength of 1610 nm, while the B12 band's central wavelength is 2190 nm [6].

In order to maximize our ability to identify methane signals from the data, we used two methods of data preprocessing. First, we created images from fractional changes in reflectance, which were calculated using a least squares estimate of both the B11 and B12 bands. The name "fractional images" refers to the fraction of B11 and B12 bands used to create the images in the formula below [11]. Here, C_s is the gradient (slope) coefficient of the least squares fit. This method is useful as it estimates methane signals from differences in the B11 and B12 reflectances from a single satellite pass.

$$\Delta B_{fractional} = \frac{C_s \cdot B_{12} - B_{11}}{B_{11}}$$

Additionally, we normalized the retrieval images. We did this because one of the major hurdles with this data was the distribution of its values. The non-zero values for the input retrieval images were

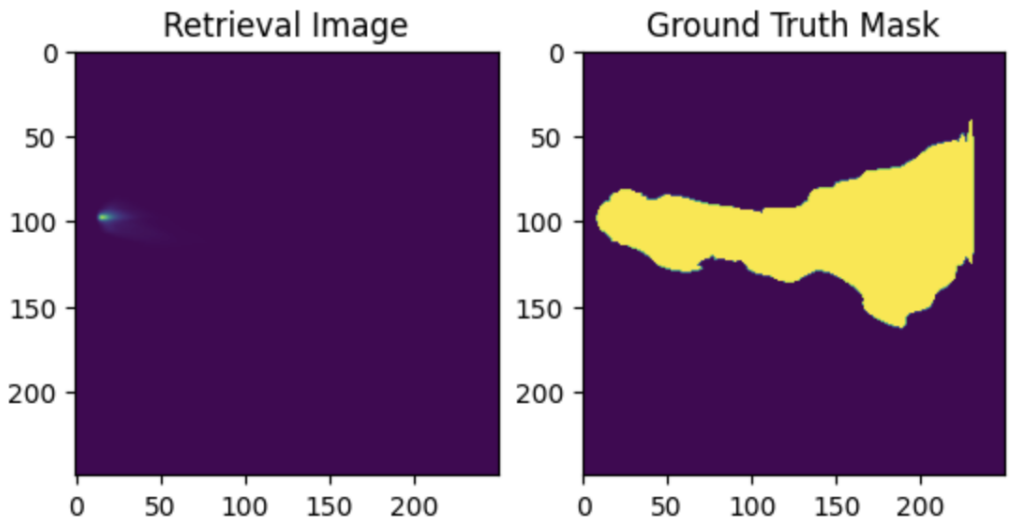

Figure 3: Example of Input Retrieval Image and Output Ground Truth Mask for a Given Asset

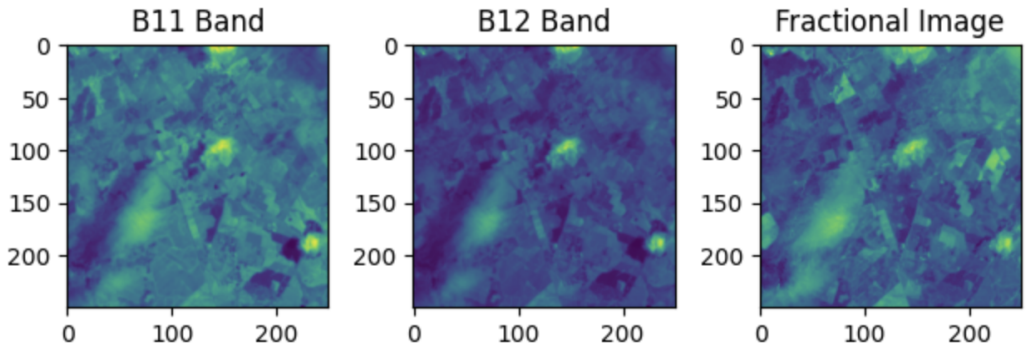

Figure 4: B11 and B12 Bands Corresponding to Above Retrieval and Mask

very clustered together (which can be seen in figure 5), making it difficult to distinguish between values that should have been masked from those that should not have. In order to spread out this distribution and gain a greater ability to differentiate between pixels that should and should not be masked, we flattened each retrieval image, took the natural log of each pixel, normalized the modified image, then reshaped the image into its original dimensions. By spreading out the distribution of the retrieval image's methane signal, we made it easier for our various models to identify pixels that should be masked.

4.2 Methodology

For this project, we tested the **hypothesis: deep learning methods will outperform traditional statistical methods for identifying methane emissions and creating methane plume masks from satellite images.**

In order to test this hypothesis, we used three primary modeling methods. First, we created a U-Net that we trained on Orbio's simulated retrieval images. Next, we used pre-trained computer vision models that specialized in pixel segmentation tasks. Finally, we compared the performance of these neural networks to a statistical threshold model.

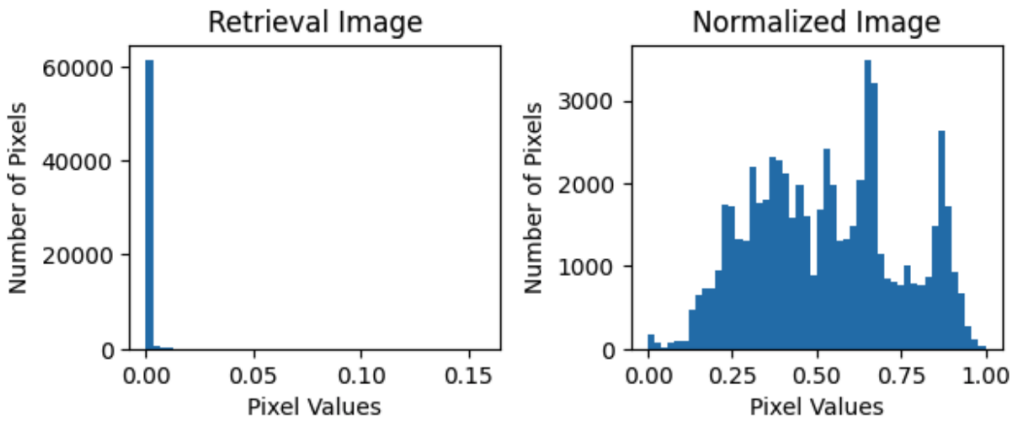

Figure 5: Histogram of Retrieval Data Before and After Normalization

We evaluated these three modeling methods using two main criteria: AUC (Area Under ROC Curve), and confusion matrix values (specifically false positive rates (FPR)). AUC was a desirable metric because of the inherent class imbalances in the ground truth masks. Additionally, we tracked false positive rates because energy companies are generally most interested in minimizing false positives. However, this comes at the cost of decreasing the model's ability to correctly predict positive values. Given this inherent trade-off, we ultimately decided to optimize for AUC. Finally, we used a representative 85/15% train/test split for our neural networks.

4.3 Results

Model	AUC	False Positive Rate
U-Net	0.501	0.50
Pre-trained ResNet (FCN)	0.502	0.32
Statistical Threshold	0.549	< 0.01

Our primary result was the success of the statistical threshold model. Although we had hoped we would surpass the baseline set by the statistical threshold model, we were impressed by its ability to create masks similar to corresponding ground truth masks. Figure 6 highlights an example of how the statistical threshold model works for a given asset. The input (the retrieval image) is top left, which is first converted to the normalized retrieval in the bottom left. Then, after applying the statistical threshold model, we are left with the statistical threshold mask in the bottom right. For this particular asset, it is evident that the predicted mask from the statistical threshold model is very similar to the ground truth plume mask in the top right.

Figure 7 highlights the hyperparameter tuning of the statistical threshold model. Specifically, we varied the value of K (the scaling factor) to see which value resulted in the highest AUC. From this hyperparameter tuning, we learned that the best-performing value of was K = 1.6.

Next, figure 8 shows a histogram of confusion matrix values calculated by using the statistical threshold model on all the assets available in the data. We learned that as we increase K in the threshold model, we improve/decrease the false positive rate, but simultaneously worsen/decrease the true positive rate.

Overall, our results highlighted just how difficult this task is, particularly due to the weak methane signals in the input retrieval images. As evidenced by the retrieval (top left) image in figure 6, many of the retrievals' methane signals are very subtle, with only a small portion of the simulated plume visible to the naked eye. This reinforces the need for normalization of the data.

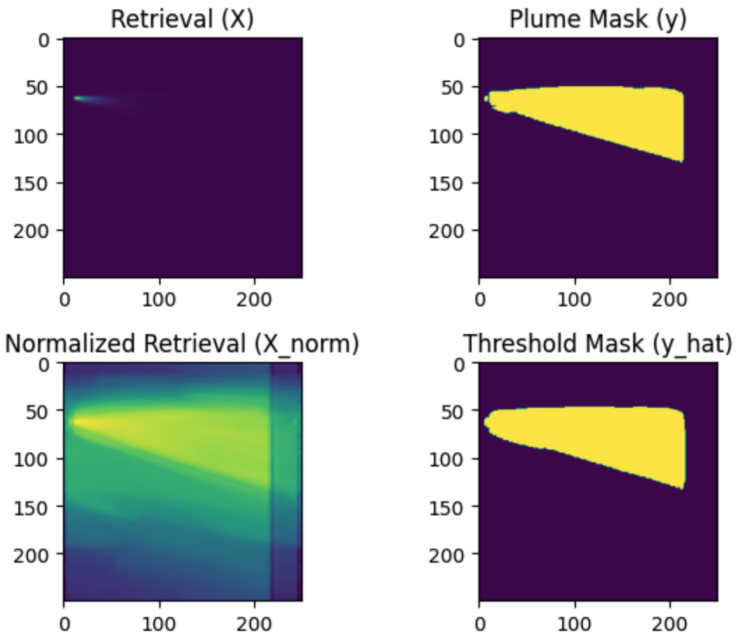

Figure 6: Example of Input Image (Retrieval) Normalized Image, Ground Truth Plume Mask, and Predicted Threshold Mask

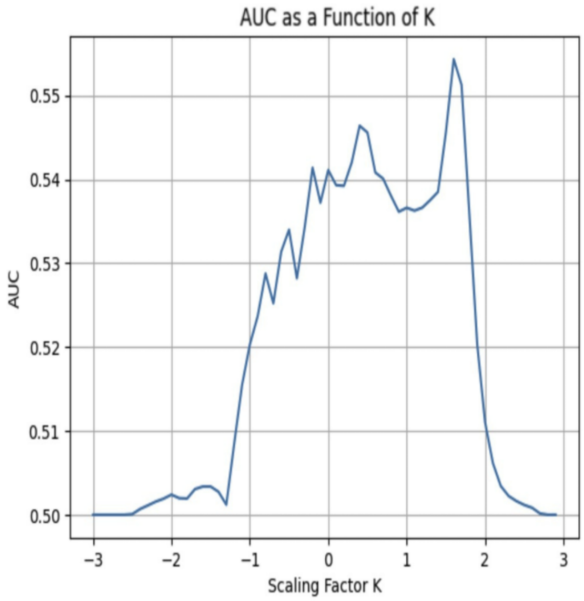

Figure 7: Hyperparameter Tuning of Statistical Threshold's Multiplication Factor K

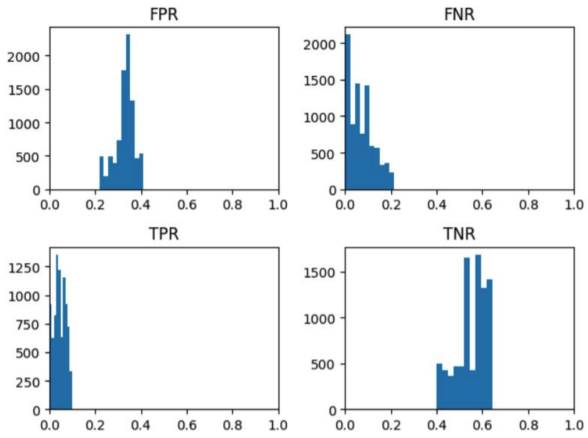

Figure 8: Histograms of Confusion Matrix Values for Statistical Threshold Model

4.4 Discussion

Ultimately, we concluded that neural networks were not the best tools for this particular task. The fine-tuned statistical threshold model performed the best, by a decent margin.

The results highlighted the fact that, although neural networks are very well suited for many other kinds of pixel segmentation problems using satellite imagery, there is much room for improvement when it comes to pixel segmentation of methane plumes. The U-Net model, despite low loss values, had difficulty identifying methane signals in both the raw and normalized inputted retrieval images. Consequently, its predicted masks rarely resembled the ground truth masks. Similarly, each of the pre-trained models worked well with more standardized images. For example, the pre-trained ResNet model worked well with segmenting animals from backgrounds. However, this ability did not extrapolate to segmenting methane plumes from simulated satellite images. Finally, the statistical threshold model, while not perfect, did have a strong ability to predict correct methane masks. Additionally, this baseline was much faster to run.

Given the time constraints of this project, we were unable to deploy our model as a cloud-based solution. However, our group is in the process of handing off our project to the Orbio team. That way, the statistical threshold model can continue to be used as a benchmark, and the U-Net model can potentially serve as a building point for future convolutional neural network solutions. We found that the main issue in this project was the strength and distribution of methane signals in the raw data. Furthermore, this task is a novel problem that remains open-ended. Consequently, because there is a limited amount of supporting research and not many existing solutions, it was inherently difficult to find alternative methods.

5 Conclusions

From our work, we concluded that traditional statistical methods were better suited to identifying methane emissions and creating methane plume masks from satellite images than deep learning methods. However, we believe deep learning architectural methods will outperform statistical methods, due to their sheer computational advantages and the availability of improved quality satellite imagery. Therefore, we are proposing our threshold model as a benchmark or reference point for upcoming architectural methods.

Given the possibility of our project being extended with the availability of resources, we identified key areas where our ideas and executions would be applicable. First, we would like to incorporate more information about the images from their metadata, such as area type, wind metrics, asset category, etc., and add these features to our modeling process to produce better results. Additionally, we could further invest resources to preprocess input images to maximize the strength of methane signals, as the dataset we worked on had generally weak methane signals.

Furthermore, we found another potentially relevant application for our statistical threshold model: cloud masking. Based on an input satellite image, the statistical threshold model was equipped to create cloud masks, with comparable AUCs to its methane masking abilities.

Another future area of work for our project would be extending our results by using transfer learning on pre-trained models. We expect to move beyond detecting methane to quantifying the presence of methane and classifying assets based on their severity levels to promote better emission standards across industries.

6 Lessons Learned

For this project, our largest problem was finding signals in the data. Even using non-noisy data, it was very difficult to draw a line between pixels that should be masked, and pixels that should not. The inherent distribution of the raw retrieval data is very clustered, and the maximum values are only slightly higher than the minimum values. Consequently, this makes it very difficult for a model to differentiate, making the creation of a methane mask very tough. Although we did not completely solve this issue, our best solution involved normalizing the data. By taking the natural log of each pixel, then normalizing each retrieval image, we were able to spread out the distribution of values, making it easier for our models to differentiate between low and high methane pixels.

Another problem for this project was the computing capabilities required to fully train our U-Net. The first issue we encountered was that our connection to AWS would crash when we attempted to train the model on the entire dataset. However, once we increased the power of the AWS instance we used, this problem was fixed. Next, we realized the substantial amount of time required to train the U-Net model on the entire dataset. To solve this issue, when tweaking minor changes to the model, we would only train the model using a smaller subset of the overall data.

Additionally, we noticed a curious disparity between the metrics and the actual images outputted by our U-Net model. Although the metrics showed that the model was decreasing in loss as it trained, and resulted in very low losses towards the end of the training, our output masks were generally very different from the ground truth masks. Once we noticed this, we realized the importance of consistently keeping an eye on model outputs in addition to metrics.

Furthermore, figuring out how to adjust the dimensions of the U-Net model was another major hurdle for this project. Generally, it was difficult to get the dimensions of the input images, linear and non-linear layers, and output layer to all work together. It was tough to understand how the various convolutions were actually changing dimensions from one step to the next. One way that we deciphered these dimensions was by learning from other examples of convolutional neural networks, specifically other U-Net models. Additionally, we found that learning more about the mathematical background for given layers in the neural network was useful for understanding how the dimensions changed. Finally, we found that it was helpful to break the problem into smaller pieces, such as isolating individual steps where dimensions changed.

In conclusion, we found two key takeaways from this project. First off, we realized that more advanced methodologies don't necessarily result in better-performing solutions. Although the pre-trained models were state-of-the-art pixel segmentation techniques, they were not trained with this particular data in mind. Additionally, despite being commonly used for pixel segmentation, the U-Net architecture we used was outperformed by our statistical threshold model. Secondly, we learned about the inherent difficulty in attempting to solve a novel problem. So far, at least in the public sphere, there is no neural network model that is capable of consistently and accurately masking methane plumes from satellite images. That said, this field is blossoming rapidly, and technological strides are occurring daily. We are excited to see how our statistical threshold benchmark contributes to the development of this field.

Student Contributions

- Alex Herron: Exploratory data analysis, data preprocessing, U-Net model, statistical threshold model, model evaluation
- Dhruv Saxena: Data preprocessing + engineering, fractional images, pre-trained model comparisons, cloud (AWS) setup and configuration

- Xiangyue Wang: Modified U-Net Model, Poster

References

[1] David T Allen, Vincent M Torres, James Thomas, David W Sullivan, Matthew Harrison, Al Hendler, Scott C Herndon, Charles E Kolb, Matthew P Fraser, A Daniel Hill, et al. Measurements of methane emissions at natural gas production sites in the united states. *Proceedings of the National Academy of Sciences*, 110(44):17768–17773, 2013.

[2] Space Studies Board, Engineering National Academies of Sciences, Medicine, et al. *Thriving on our changing planet: A decadal strategy for Earth observation from space*. National Academies Press, 2019.

[3] Ocko et al. Acting rapidly to deploy readily available methane mitigation measures by sector can immediately slow global warming. *Environmental Research Letters*, 16(5):054042, 2021.

[4] Ronneberger et al. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[5] Veronika Eyring, NP Gillett, Krishna Achutarao, Rondrotiana Barimalala, Marcelo Barreiro Parrillo, Nicolas Bellouin, Christophe Cassou, Paul Durack, Yu Kosaka, Shayne McGregor, et al. Human influence on the climate system: Contribution of working group i to the sixth assessment report of the intergovernmental panel on climate change. *IPCC Sixth Assessment Report*, 2021.

[6] GISGeography. Sentinel 2 bands and combinations, Jun 2022.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] International Energy Agency (IEA). Global methane tracker 2022. 2022.

[9] Siraput Jongaramrungruang, Andrew K Thorpe, Georgios Matheou, and Christian Frankenberg. Methanet–an ai-driven approach to quantifying methane point-source emission from high-resolution 2-d plume imagery. *Remote Sensing of Environment*, 269:112809, 2022.

[10] AK Thorpe, C Frankenberg, and DA Roberts. Retrieval techniques for airborne imaging of methane concentrations using high spatial and moderate spectral resolution: Application to aviris. *Atmospheric Measurement Techniques*, 7(2):491–506, 2014.

[11] Daniel J Varon, Dylan Jervis, Jason McKeever, Ian Spence, David Gains, and Daniel J Jacob. High-frequency monitoring of anomalous methane point sources with multispectral sentinel-2 satellite observations. *Atmospheric Measurement Techniques*, 14(4):2771–2785, 2021.

[12] Steven C Wofsy and Steve Hamburg. Methanesat-a new observing platform for high resolution measurements of methane and carbon dioxide. In *AGU Fall Meeting Abstracts*, volume 2019, pages A53F–02, 2019.