Classifying Methane Emission Sources From Publicly Available Satellite Imagery

Kelechi Uhegbu
Department of Computer Science
Stanford University
kuhegbu@stanford.edu

William Zhang
Department of Computer Science
Stanford University
wxyz@stanford.edu

Abstract

Methane emissions mapping is currently a labor intensive process, but advances in remote sensing and computer vision present the possibility of automating this process. We apply computer vision techniques towards multi-class classification of methane emissions sources depicted in satellite imagery. We use a location dataset of labeled methane emitting facilities and gather a new image dataset from publicly available sources. We implement baseline models and fine-tune ResNet and Vision Transformer models for this task. We explore various methods of data augmentation, hyperparameters, and optimization methods to increase classification accuracy. We find that fine-tuned ResNet-50 performs the best on our task, due to its ability to recognize small identifying features in the satellite imagery. We hope that our work will aid in the development of a automated and scalable system for identifying and classifying methane emitting facilities.

1. Introduction

Methane is the second-largest contributor to the greenhouse effect and its concentration in the atmosphere is currently increasing at a rate of around 1% per year. Methane is a more potent greenhouse gas than carbon dioxide, and is a major contributor to climate change. [5] Currently, multiple global mapping satellite missions such as SCIAMACHY, GOSAT, and TROPOMI provide atmospheric methane concentrations data. [11] These allow researchers to map the concentration methane emissions using satellite imagery [9], but it is difficult to classify the sources of methane emissions from satellite imagery. An automated and scalable classification system would be instrumental to attributing methane emissions detected by these satellites to facilities on the ground. This problem is important for carbon accounting, emissions surveys, and environmental regulation. We aim to leverage computer vision techniques to classify facilities that are known to emit methane from publicly available satellite imagery.

A recent article highlights the rise in satellite-based methane detection methods and discusses the implications of this on policy, industry, and finance. [5] Quantifying emissions from space is a recent development, which has advantages over ground-based monitoring. Ground based sensors offer more accurate data, but it is localized and requires an upfront investment. In contrast, remote sensing can provide data about anywhere in the world, and it is much cheaper to obtain.

Recently, the California Air Resources Board collaborated with NASA to conduct the California Methane Survey, which aimed to identify and classify large methane plumes across California. This project surveyed over 10,000 square miles, which identified 564 methane emitting facilities. [2] However, the identification process was performed manually and this project took over 2 years to complete, and the researchers highlight the need for consistent and continuous observations of methane emitting facilities. [18]

This vast amount of satellite imagery makes human classification infeasible, which is why we aim to automate this task using computer vision methods. We hope that this enables methane emission source classification to be highly scalable across the world to allow for continuous observation and identification of methane emission sources.

Using satellite imagery for classification comes with limitations. Satellite imagery provides a top down view of the world from far away, which means that the input data will be limited in its spatial resolution [1]. Low spatial resolution results in increased difficulty in distinguishing between objects, which is problem for classification tasks.

1.1. Problem Statement

This problem is formulated as a multi-class classification of satellite images. Given a latitude and a longitude of a location of a methane emitting facility, we collect a satellite image of the surrounding region and predict its class.

For our project, we use a dataset with locations and classes of methane emissions. We gather satellite imagery corresponding to these locations from public sources which is input into a image classification model. The model pre-

dicts the class of the facility corresponding to the location.

We fine-tune ResNet-50 and Vision Transformer (ViT) models and compare performance against several baseline classification methods. We also test various methods of data augmentation. The fine-tuned ResNet-50 model attains a maximum of 68% accuracy on the validation data. We conduct an analysis of the quantitative and qualitative results, including accuracy, confusion matrices, saliency maps, and common errors.

2. Related Works

One related work has explored classification of methane emitting lakes. [14] However, they do not use computer vision methods and rely on manual labelling. Our problem also focuses on human infrastructure as the source of methane emissions, rather than natural sources.

Another work analyzes satellite imagery to find point sources of industrial methane emissions. [10] These researchers were able to identify a few dozen facilities that were emitting extreme amounts of methane due to inefficient flaring operations. However, they use manual labelling to classify these facilities. Our work aims to automate the task of classifying the emissions sources, which will allow our method to be easily scalable across the entire world.

The California Methane Survey uses satellite based imaging of methane plumes to identify methane emitting facilities across 10,000 square miles of California, and demonstrated the use case of remote sensing to rapidly and repeatedly assess large areas for an unclassified population of methane point sources. [18] This work also uses manual labelling for classification, which was done over the span of 2 years. [2] This highlights the need for an automated method for classifying these emission sources, which our work aims to address. This work also does not solely rely on satellite imagery, as the researchers also used mapping software, location data, and other resources to determine what these facilities were.

Some researchers have begun using satellite imagery to create a database of oil and gas infrastructure. [19] Their model (OGNet) outputs the probability that a given satellite imagery tile contains an oil and gas facility. This is used for binary classification of satellite images into tiles that do contain oil and gas infrastructure and those that do not. However, their dataset only contains 149 images of oil refineries and 6,917 negative images, while our dataset contains many more images of facilities that emit methane. Our work also presumes that the satellite images we are analyzing contain an image of a methane emitting facility, since we are able to map locations of methane emissions using satellites already. [9] We also perform multi-class classification of many more types of methane emitting facilities, rather than just binary classification of oil and gas infrastructure.

Several datasets and challenges are focused on automated classification and detection of human infrastructure in satellite imagery, such as Functional Map of the World (fMoW) [3] and xView [12]. Other work has focused on deep learning methods to map solar photovoltaic panels [22] and wind turbines [23] from satellite imagery. However, none of these focus on classification of methane emitting infrastructure specifically. Some, like xView, are also formulated as object detection tasks rather than image classification tasks. We can use insights gained from these challenges and their solutions to improve our model.

3. Dataset and Features

We use the "Sources of Methane Emissions (Vista-CA)" dataset, which was created by NASA under the North American Carbon Program. [8, 17] The dataset contains over 230,000 locations of methane emission sources in California, along with a label for what type of facility is at each location. The different classes are as follows: ['Composting Sites', 'Dairies', 'Digesters', 'Feed Lots', 'Landfills', 'NG Fueling Stations', 'NG Stations', 'Oil and Gas Facilities', 'Oil and Gas Field Boundaries', 'Oil and Gas Wells', 'Power Plants', 'Processing Plants', 'Refineries', 'Storage Fields', 'Wastewater Treatment Plants']. However, there is a massive class imbalance in this dataset, with almost 200,000 examples of 'Oil and Gas Wells'. To mitigate this, we take a maximum of 1000 examples from each class and use that as our dataset. This leaves us with 5938 datapoints.

We corresponding satellite imagery satellite imagery published under the National Agriculture Imagery Program (NAIP) from 2016 to 2016, with resolution of 1 m/pixel. [8] The images were processed and downloaded using Google Earth Engine. [6] This imagery do not suffer from cloud cover or haze because images were acquired on days with low cloud cover. Each image contains RGB bands, and captures a 500 m $\times$ 500 m area centered at the coordinate of the corresponding location in the dataset. However, some of the coordinates are not placed exactly at the center of the facility, so they may only appear at the edge of the image.

We randomly select 20% of the datapoints for a validation set, and use the remaining 80% for training.

4. Methods

Using pytorch [15] and scikit-learn [16], we implement various models to train on the satellite image dataset. Various methods include:

- Naive Bayes Classifier
- Support Vector Machine Classifier
- ResNet-50
- Vision Transformer (ViT)

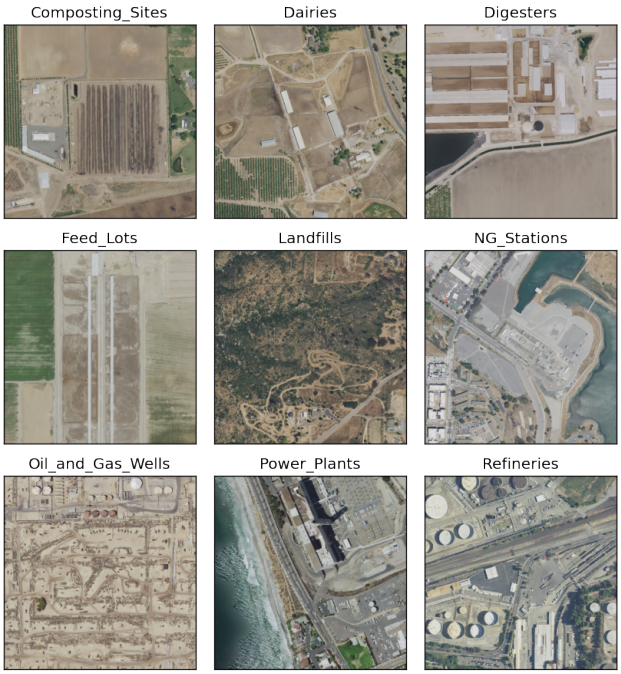

Figure 1. Example images from a few of the classes in the dataset.

4.1. Baseline Methods

We use a Naive Bayes classifier and a Support Vector Machine classifier as our baseline models. Naive Bayes is a rudimentary yet computationally efficient probabilistic model that works better than random guesses, so it is useful as a initial baseline. Naive Bayes chooses the class satisfying the equation:

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y).$$

For our purposes, x_i represents each pixel, and $\hat{y}$ is the predicted label. The Naive Bayes algorithm assumes that each pixel is independent, which is an incorrect assumption. We expect this model to perform relatively poorly.

Support Vector Machine (SVM) works to create very distinct decision boundary between classes. This could be visualized as a hyperplane to separate images corresponding to their class. Since our data is very high dimensional, a fully trained SVM model will attain very close to 100 percent accuracy for the training data, but not necessarily the test data. We use the scikit-learn implementations of these models.

4.2. Deep Learning Methods

We fine-tune a ResNet-50 model pretrained on the ImageNet dataset on our training data for 20 epochs. ResNet-50 is a popular deep learning residual model used for image classification [7]. For our classification task, we replace the last linear layer to one with 15 nodes.

We also fine-tune Vision Transformers [4], which are a new technique that is advertised that it can attain excellent results compared to convolutional networks while requiring fewer computational resources to train. ViT adapts transformer methods which were used in NLP for computer vision tasks. Instead, ViT uses 16x16 pixel patches to feed into its attention heads. Similar to ResNet, we replaced the head with 15 nodes for our downstream task.

For these models, we use the torchvision implementation. [13]

We optimize these methods to minimize cross entropy loss, which can be represented by this equation:

$$L = -\frac{1}{N} \sum_{i} y_i \dot{\log}(\hat{y}_i)$$

where N is our total number of samples, y is the vector of true labels for the sample (in our case, one hot vector for our classes), y_i is our predicted probabilities for that sample.

5. Experiments

For SVM and Naive Bayes, we use the default configurations specified by scikit-learn. We do not perform any pre-processing or feature extraction.

For the deep learning models, we use a batch size of 32, and use cross entropy loss as the loss function. Our batch size was chosen using a validation set, and was also impacted by the amont of GPU memory available.

We chose to use ResNet-50 and ViT-B/16 pretrained on ImageNet. We also tried these models without pretraining, but stopped these experiments early as their performance was far worse.

ResNet-50 takes a 224 $\times$ 224 pixel image as input for data. We experimented with several methods of preprocessing to make the input size of the image match the input size of ResNet: center cropping the image to 224 $\times$ 224 pixels and resizing the image to 224 $\times$ 224 pixels.

As we observed a high degree of overfitting in early experiments, we also experimented with several image transforms in the training phase, specifically random cropping and vertical and horizontal flipping.

We train the ResNet models using SGD optimizer with learning rate 1×10^{-3}, momentum 0.9. We also ran experiments where we trained ResNet using Adam optimizer, but they yielded worse results so we do not report the results here. SGD was also the optimizer used in the original ResNet paper [7], and supporting literature shows that in an over-parameterized setting, adaptive optimization methods like AdaGrad, RMSProp, and Adam do not generalize as well as SGD. [21] We believe that this applies to our situation since ResNet-50 has a high number of parameters and we are only doing classification for 15 classes.

For Vision Transformers, we also used a 224 $\times$ 224 pixel image as input. We used a combination of random cropping and vertical and horizontal flipping during the training phase, since that yielded the best results in our ResNet-50 experiments. We used the Adam optimizer which was shown to perform the best in the original paper. [4]

To compare results, we used final training and validation accuracy. This was computed as the percentage of correctly labeled images using the highest probability outputted by the model.

6. Results and Discussion

Our results are seen in Table 1.

Table 1. Results for Classification Models

Model	Training Acc	Validation Acc
Naive Bayes	0.2378	0.2205
SVM	0.9915	0.3047
ResNet50 - Cropped	0.9385	0.6305
ResNet50 - Resized	0.9242	0.6069
ResNet50 - Rand Crop	0.8432	0.6571
ResNet50 - Rand Crop + Flip	0.8276	0.6838
ViT-B/16	0.4817	0.4857

As expected, SVM attains near perfect training accuracy but relatively low validation accuracy. Naive Bayes does not perform well on this task. The fine-tuned ResNet-50 models perform the best, with the best validation accuracy around 68.3%. The ResNet-50 models do show a high degree of overfitting, which is expected given the small size of our dataset. ViT-B/16 performs relatively poorly, but interestingly did not exhibit overfitting.

It is worth noting that this task is also very difficult for humans to perform if only given the satellite imagery, so it is impressive that our best model was able to correctly classify a majority of the images. Many of the classes are hard to distinguish, as they do not all have distinctive features that separate one from another. This is in contrast to datasets like CIFAR-10, where each class is very distinct from the others. Therefore, the accuracy that our model is able to achieve is quite good considering the difficulty of the task.

One surprising result was the poor accuracy of the Vision Transformer model. It took on average 70 minutes to run, yet had lower accuracy than ResNet50. We believe this is due to the small features that are present in satellite imagery in addition to the class disparity of our dataset. Transformers work well when the image patches have some sort of sequential structure, where features on one patch may continue into another. However, in our satellite images, the features that determine class, such as buildings, storage tanks, and cows, are small enough that they fit into one patch. This leads us to believe that the attention mechanism is not particularly helpful in this application since the sequential structure of the data is not as prominent.

6.1. Confusion Matrix

6.1.1 ResNet

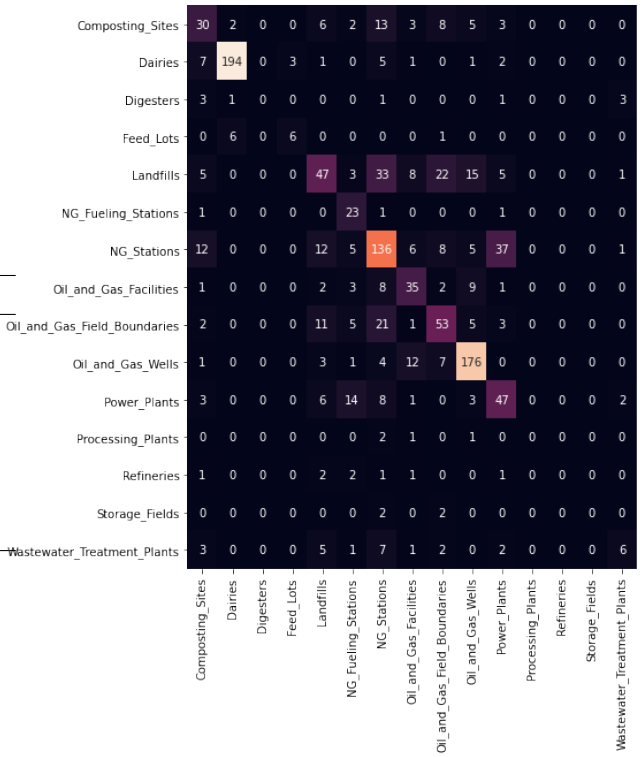

Figure 2. Confusion matrix of the ResNet model for the validation data. The true class appears on the y-axis, and the predicted class is on the x-axis.

We created a confusion matrix for our best performing model to analyze the most common errors it made. As we expected, the common mistakes that the model made were also ones that humans would make. For example, NG Stations, NG Fueling Stations, and Power Plants were often mistaken for each other. Upon inspection of the images, these classes have many landmarks in common, such as buildings, storage tanks, and roads. Therefore, it seems that our model is also learning these landmarks as a way to classify the images.

The model performed particularly well on classifying Dairies and Oil and Gas Wells. This makes sense because both of these classes have unique identifying landmarks that none of the other classes possess. Images in the Dairies class contain cows, while images in the Oil and Gas Wells class contain oil and gas drills. This makes it easy even for humans to classify these images, so it is no surprise that the model is also able to do well on these.

It seems that the model had a difficult time classifying

Landfills, as it would often mistake landfills for many other classes. Upon inspection of the data, it is apparent that landfills can vary widely, appearing in many sizes, shapes, and textures. Landfills do not have any sort of unique identifying landmarks, and may appear near homes, roads, and other landmarks commonly seen in other classes. The landfills in this dataset were not limited to landfills currently in use, so some of the landfills may have been covered over with dirt and vegetation. Therefore, it is difficult for even a human to distinguish whether or not an image is a landfill.

6.1.2 Vision Transformer

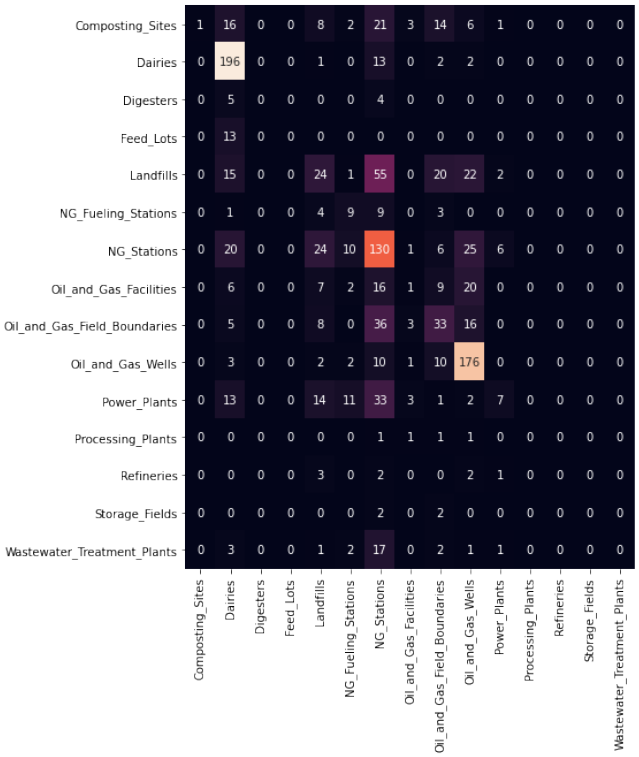

Figure 3. Confusion matrix of the Vision Transformer model for the validation data. The true class appears on the y-axis, and the predicted class is on the x-axis.

For the Vision Transformer model, it performs best on images for Dairies, Oil and Gas Wells. Similarly to the ResNet model, the Vision Transformer was able to find the unique identifying features of these images.

One large difference in comparison to the ResNet model was how they identified Wastewater_Treatment_Plants. There were not many samples to begin with, but the ResNet model was still able to identify some of those images correctly. However, the Vision Transformer model identifies none of these labels correctly.

6.2. Saliency Maps

We also create saliency maps to visualize the importance the model puts on the individual pixels of an image to make its prediction. [20] Saliency maps are beneficial for gaining insights on what a model pays attention to, and are created by using gradient ascent to maximize the class score for an image. For our case, we use the image-specific approach, where we input the image and the correct class to get the saliency map for that image.

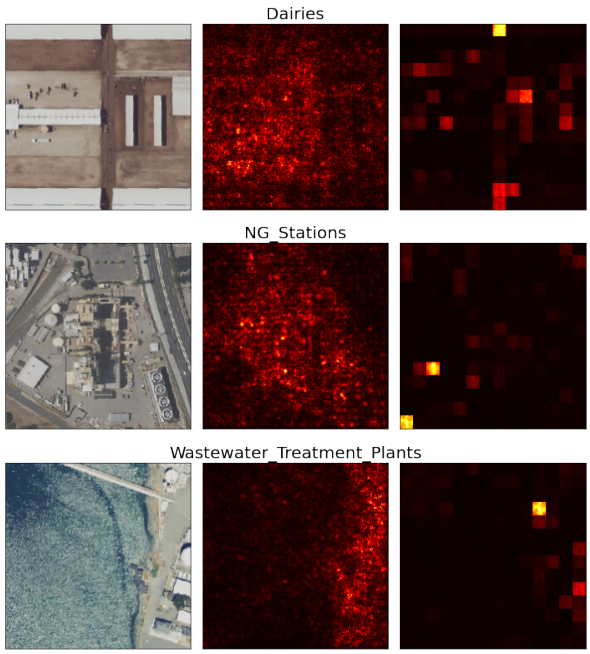

Figure 4. Example Saliency maps for validation images for ResNet and ViT in that order.

As seen in Figure 4, it seems that the main reason why ResNet outperformed Vision Transformers was due to the fact that the size of the identifying features for each is too small for the Vision Transformers model.

In the first example, this is a dairy farm with some number of cows. The ResNet model does pay attention to the areas with cows, but the ViT model does not pay attention to that region.

In the second example for NG_Stations, the Vision Transformer model pays attention to the lower right areas. These are not too important in terms of identifying features. However, the ResNet model pays attention to the central area of the image, which includes the main part of the station.

For the third example for Wastewater_Treatment_Plants, the ResNet correctly pays attention to the right side of the image, which is the treatment plant and the surrounding area itself instead of the water. On the other hand, the Vision Transformer model pays attention to the area where the bridge meets the land.

Overall, the Vision Transfomer model is unable to pay attention to the smaller features in the images due to the patch size being larger than many of the defining landmarks of each class. Due to the small identifying features for satellite imagery, ResNet is more suited for the task.

6.3. Qualitative Results

We use softmax on the logits to get a probability distribution over the classes for the images in the validation set. This allows us to identify common errors and assess the confidence of the model in its predictions.

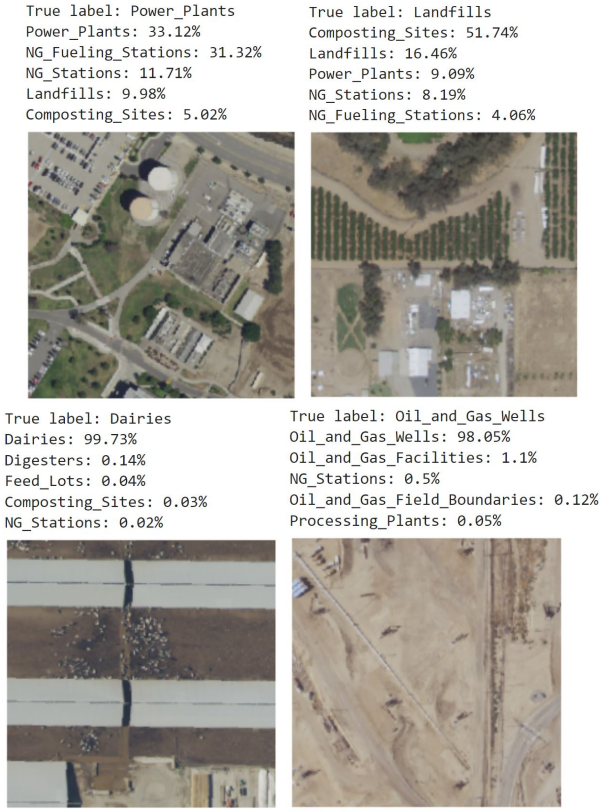

Figure 5. True labels and predictions with probabilities for a few example images in the test set using the best ResNet model.

As we can see in Figure 5, the model ranges from not very confident about its prediction for the images containing Power_Pants and Landfills, to very confident about the predictions for Dairies and Oil_and_Gas_Wells. We believe that this occurs due to the diversity of the images in each class. Since most images for Dairies look similar to one another, in that they have a large building and cows (which do not appear in most other classes), if the model detects these landmarks, then it can be very confident that the image is in the Dairies class. The same can be said for Oil_and_Gas_Wells, since these images contain many oil and gas drills, which usually do not appear in other classes.

For the Power_Pants and Landfills classes, these appear in many different regions and have a much more varied appearance. In addition, there are many types of power plants and landfills which further diversifies the classes, and they do not have very specific identifying features, making the model much less confident about its predictions for these.

We also see that for the Power_Plants image, the model also outputs a high probability for NG_Fueling_Stations, likely due to the storage tanks that appear at the top center of the image. For the Landfills image, the model predicts Composting_Sites as the most likely class, due to the presence of vegetation, dirt, and buildings, which are also identifying features for that class. In fact, most images of Landfills do not contain commercial buildings, which makes this image different, leading to the incorrect classification.

Many other common errors such as these can be seen. Many of the Natural Gas and Oil and Gas facilities are confused with each other due to the presence of common infrastructure and landmarks. These facilities are also often located near one another, leading to the common buildings and the ground texture. On the other hand, Feed_Lots are often confused for Dairies, due to the presence of cows in Feed Lots. However, the number of examples of Dairies is higher, so the model classifies these images as Dairies instead.

7. Conclusion and Future Work

Our work demonstrates that methane emitting facilities can be classified using deep learning methods, which paves the way for future works that can perform this task on a global scale. We tested ResNet and Vision Transformer models and compared them to baseline models and found that ResNet is more suited for this task. The small size of identifying features is too small for the 16x16 patches that the Vision Transformer model uses. On the other hand, ResNet is still able to pay attention to these smaller features.

Due to time and computation constraints, we trained our deep learning models for 20 epochs. If given more time, we would experiment by changing our learning rates and number of epochs to determine the best setting for these hyperparameters.

In addition, we worked on a dataset with methane emitting facilities. In the future, we would like to create an end-to-end model that uses a methane concentration map to select locations with high emissions and then classify them. It would also be interesting to see how auxiliary information (such as location, methane concentration, and zoning) may impact the performance of the model.

We hope that our work will aid in the development of a automated and scalable system for identifying and classifying methane emitting facilities, and that it will aid in efforts to decrease methane emissions through carbon accounting and environmental regulation.

8. Contributions

WZ: Gathered the location dataset. Wrote the code to export the image dataset from Google Earth Engine. Implemented dataloader and model training loop in pytorch, and ran the ResNet experiments. Created visualizations for confusion matrix, predictions, and probabilities.

KU: Set up AWS for experiments. Implemented baseline models (SVM and Naive Bayes). Ran ViT experiments, and generated saliency maps.

This code was used as foundation to generate saliency maps: `https://github.com/sunnynevarekar/pytorch-saliency-maps/blob/master/Saliency_maps_in_pytorch.ipynb`

References

[1] F. A. Al-Wassai and N. Kalyankar. Major limitations of satellite images. *arXiv preprint arXiv:1307.2434*, 2013.

[2] C. A. R. Board. California methane surveys, Jun 2020.

[3] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee. Functional map of the world. In *CVPR*, 2018.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[5] J. Elkind et al. Nowhere to hide: Implications for policy, industry, and finance of satellite-based methane detection. *Columbia School of Public and International Affairs Center on Global Energy Policy*, 2020.

[6] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[8] F. Hopkins, T. Rafiq, and R. Duren. Sources of methane emissions (vista-ca), state of california, usa, Dec 2019.

[9] E. S. Imaging. Mapping methane with vhr swir imagery, Jun 2021.

[10] I. Irakulis-Loitxate, L. Guanter, Y.-N. Liu, D. J. Varon, J. D. Maasakkers, Y. Zhang, A. Chulakadabba, S. C. Wofsy, A. K. Thorpe, R. M. Duren, et al. Satellite-based survey of extreme methane emissions in the permian basin. *Science Advances*, 7(27):eabf4507, 2021.

[11] D. J. Jacob, A. J. Turner, J. D. Maasakkers, J. Sheng, K. Sun, X. Liu, K. Chance, I. Aben, J. McKeever, and C. Frankenberg. Satellite observations of atmospheric methane and their value for quantifying methane emissions. *Atmospheric Chemistry and Physics*, 16(22):14371–14396, 2016.

[12] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018.

[13] S. Marcel and Y. Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488, 2010.

[14] E. Matthews, M. S. Johnson, V. Genovese, J. Du, and D. Bastviken. Methane emission from high latitude lakes: methane-centric lake classification and satellite-driven annual cycle of emissions. *Scientific Reports*, 10(1):1–9, 2020.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] T. Rafiq, R. M. Duren, A. K. Thorpe, K. Foster, R. Patarsuk, C. E. Miller, and F. M. Hopkins. Attribution of methane point source emissions using airborne imaging spectroscopy and the vista-california methane infrastructure dataset. *Environmental Research Letters*, 15(12):124001, 2020.

[18] I. M. Riley Duren, Andrew Thorpe. The california methane survey, 2020.

[19] H. Sheng, J. Irvin, S. Munukutla, S. Zhang, C. Cross, K. Story, R. Rustowicz, C. Elsworth, Z. Yang, M. Omara, et al. Ognet: Towards a global oil and gas infrastructure database using deep learning on remotely sensed imagery. *arXiv preprint arXiv:2011.07227*, 2020.

[20] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013.

[21] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017.

[22] J. Yu, Z. Wang, A. Majumdar, and R. Rajagopal. Deepsolar: A machine learning framework to efficiently construct a solar deployment database in the united states. *Joule*, 2(12):2605–2617, 2018.

[23] S. Zhou, J. Irvin, Z. Wang, E. Zhang, J. Aljubran, W. Deadrick, R. Rajagopal, and A. Ng. Deepwind: Weakly supervised localization of wind turbines in satellite imagery. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.