LUND UNIVERSITY

Classification of aerosols in CALIOP data using unsupervised semantic segmentation

Author: Sixten Rosager

Thesis submitted for the degree of
Bachelor of Science
Project duration: Three Months
Supervisor: Johan Friberg
Examiner: Edouard Berrocal

Department of Physics
Division of Combustion Physics
May 2024

Abstract

This thesis investigates the feasibility and efficacy of employing unsupervised semantic segmentation for classifying features in CALIOP data, aiming to address significant bias inherent in current classification methods. By exploring various preprocessing techniques, dimensionality reduction methods, and classification algorithms, the study evaluates the potential of semantic segmentation in improving the accuracy of aerosol classification. Despite computational limitations imposed by working on a standard laptop, the research produces promising results, demonstrating the capability of certain model configurations to identify important features and maintain continuity in segmented images. The implications of mitigating bias in CALIOP data are profound, with potential improvements in understanding aerosol radiative forcing and enhancing climate model predictions. While this study represents a significant step towards replacing current classification methods like SIBYL, further research is warranted to explore and refine the approaches introduced here.

Contents

Abbreviations

AI : Artificial Intelligence

CALIOP : Cloud-Aerosol Lidar with Orthogonal Polarization

CALIPSO : Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations

CCA : Connected Component Analysis

CNN : Convolutional Neural Network

DBSCAN : Density-Based Spatial Clustering of Applications with Noise

LiDAR : Light Detection And Ranging

NASA : National Aeronautics and Space Administration

PCA : Principal Component Analysis

ReLU : Rectified Linear Unit

RMS : Root Mean Square

SGD : Stochastic Gradient Descent

SIBYL : Selective Iterated BoundarY Location

SAA : South Atlantic Anomaly

t-SNE : t-distributed Stochastic Neighbor Embedding

1 Introduction

Volcanic eruptions and large forest fires inject substantial amounts of aerosols into the stratosphere, influencing atmospheric dynamics and climate patterns. NASA's CALIOP LiDAR instrument plays a crucial role in quantifying atmospheric aerosols, providing valuable data for climate projections. However, this is a difficult task. Especially when it comes to differentiating between smoke layers and ice clouds in the stratosphere. This can be done qualitatively by looking at the depolarisation ratios of the structures but the automated layer detection algorithm SIBYL (Selective Iterated BoundarY Location) struggles. This results in the accuracy of CALIOP data, particularly in calculating optical depth, becoming compromised with significant bias. The magnitude of the bias depends on various factors, including latitude and altitude.

Kar et al.[1] compares optical depth measurements from CALIOP and SAGE III across latitudes, revealing bias below 10% near the equator but soaring to 100% at mid-latitudes. This bias stems not from the data itself but from the SIBYL algorithm struggling to cope with the data's noise level. As a result, classifications may display artefacts diverging from physical reality. Accurate quantification of aerosol optical properties and their distribution is vital for climate projections, aiding in understanding radiative forcing and predicting future climate scenarios.

To address these challenges, this thesis proposes the application of unsupervised semantic segmentation, a cutting-edge computer vision technique leveraging convolutional neural networks (CNNs), to classify different structures in CALIOP data. Unlike traditional threshold-based approaches, semantic segmentation algorithms like CNNs can identify and segment aerosol structures based on their features, potentially reducing bias and improving classification accuracy. The objective is to develop a model capable of reliably discerning between smoke and ice clouds.

Current approaches to addressing bias in CALIOP data, such as adjusting optical depth values to match other observations, are rudimentary and fail to account for regional or aerosol-specific variations in bias. By contrast, unsupervised semantic segmentation offers a more nuanced and adaptive solution, which could significantly improve our understanding of aerosol radiative forcing and, consequently, climate modelling accuracy.

This thesis aims to demonstrate the feasibility and effectiveness of unsupervised semantic segmentation in classifying features in CALIOP data. The model will be trained on data spanning from December 2019 to March 2020, a period characterized by significant forest fires. The research methodology encompasses various preprocessing techniques, dimensionality reduction methods, and hyperparameter optimisation to optimise classification performance. Evaluation will primarily be qualitative, assessing the quality of segmented images produced by different approaches relative to each other.

2 Theory

2.1 The Stratosphere & Radiative Forcings

The stratosphere is a layer of Earth's atmosphere situated above the troposphere, extending from about 10 to 50 kilometres above the Earth's surface. It plays a crucial role in regulating the planet's climate and weather patterns. Unlike the troposphere below it, where temperature generally decreases with altitude, the temperature in the stratosphere increases with altitude due to the absorption of solar radiation by ozone molecules.

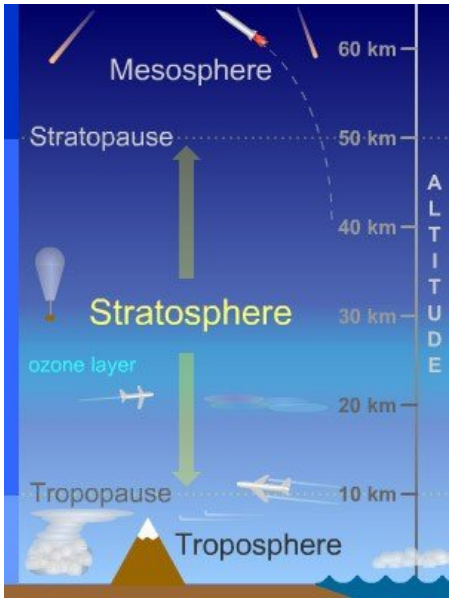

Figure 1: Schematic image of the stratoshere taken form a paper by Bauk et al.[2]

One of the key factors influencing the climate dynamics of the stratosphere is the presence of aerosols, tiny solid or liquid particles suspended in the air. These aerosols can be natural, such as volcanic ash and sea salt, or human-made, like those produced by industrial activities. Aerosols can have substantial impacts on Earth's climate by affecting the balance of incoming solar radiation and outgoing thermal radiation, a phenomenon known as radiative forcing.

Stratospheric aerosols are crucial for several reasons, most notably due to their longer lifetimes compared to lower atmospheric aerosols, allowing them more time to interact with incoming solar radiation. Additionally, stratospheric aerosols can be transported over long distances by atmospheric circulation patterns, leading to widespread effects on climate. Moreover, certain types of aerosols, such as sulfate particles which are created by sulfur dioxide emitted by volcanic eruptions, can have particularly potent cooling effects in the stratosphere by reflecting sunlight back into space, contributing to a net decrease in global temperatures. Therefore, while stratospheric aerosols represent only a small fraction of atmospheric constituents, their radiative forcings can have significant implications for Earth's climate system.

2.2 The CALIOP instrument

CALIOP, short for Cloud-Aerosol Lidar with Orthogonal Polarization, was an elastic backscatter LiDAR instrument employed aboard the CALIPSO satellite[3]. CALIPSO, which stands for Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observation, was a component of NASA's CNES's joint venture initiative to study the global radiative impacts of aerosols and clouds on climate. Launched in 2006, CALIPSO provided continuous measurements of clouds and aerosols until the conclusion of its mission in 2023.

CALIPSO operated at an altitude of approximately 685 kilometres above the Earth's

surface and orbited the Earth in a polar orbit. The satellite completed almost 15 orbits around the Earth per day, with an orbital period of about 98 minutes. It captured comprehensive atmospheric data spanning different latitudes and longitudes. An illustration of these orbits can be seen in Figure 2.

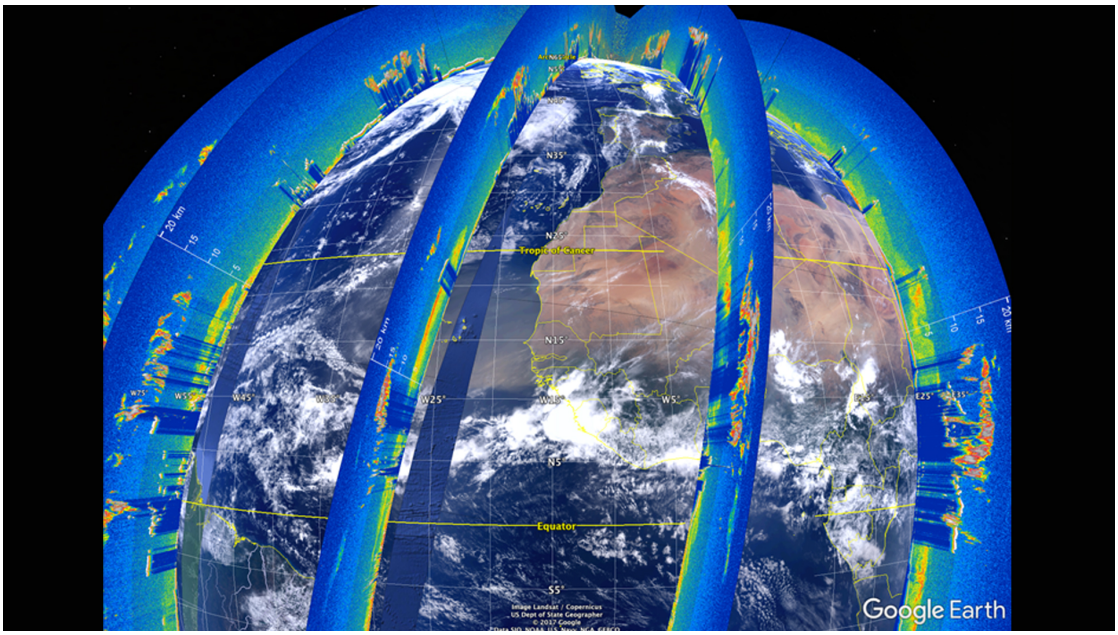

Figure 2: CALIOP observations of lidar backscatter superimposed on a weather map. Image credits go to NASA DEVELOP and the image was taken from: `https://appliedsciences.nasa.gov/what-we-do/projects/enhancements_to_visualization_of_calipso_vocal_through_case_studies_of_saharan_dust`

The CALIOP instrument emits laser pulses of two different wavelengths 1064 nm, and 532 nm. These pulses are transmitted downward toward the Earth's surface and are scattered back to the instrument by clouds and aerosols. The laser beam emitted by CALIOP has a narrow width, typically around 70 meters at the Earth's surface, enabling precise vertical profiling of clouds and aerosols in the atmosphere. The energy of the pulses needs to be high enough for them to be able to penetrate through atmospheric scattering, detect weak return signals from distant targets, and achieve accurate and reliable measurements of the Earth's surface or atmosphere.

The distance from the satellite to the surface is long, and the pulse is short, resulting in energy levels on the order of millijoules. Consequently, there is a significant amount of noise, particularly noticeable in the 1064 nm channel. This channel, with its lower energy and longer wavelength, experiences only 1/16th as much Rayleigh scattering. Additionally, for the majority of particle sizes, it experiences negligible Mie scattering.

However, the challenges are exacerbated during daytime operation. Increased solar radiation intensifies noise levels by interfering with the LiDAR signal. Furthermore, daytime atmospheric turbulence can scatter and refract the laser beam, leading to fluctuations and distortions in the received signal, which further contribute to higher levels of noise in the LiDAR data.

When producing the data different resolutions were chosen for different altitudes. This is necessary since the density of the air decreases exponentially with increasing altitude. A lower density means less backscattering which in turn means a weaker signal. Additionally, at lower altitudes, the laser pulse has to penetrate through more atmospheric layers, resulting in greater attenuation and reduced signal strength. In both cases, the weaker signal leads to decreased sensitivity and accuracy in detecting aerosols and clouds compared to higher-altitude measurements.

By analyzing the characteristics of the backscattered light, CALIOP provides valuable insights into the vertical distribution, composition, and optical properties of clouds and aerosols, contributing to our understanding of Earth's atmosphere, climate, and air quality.

Figure 13 in the paper "CALIPSO level 3 stratospheric aerosol profile product: version 1.00" by Kar et al.[1] compares the atmospheric optical depth calculated using aerosol measurements from CALIOP and SAGE III. Atmospheric optical depth is a measure of the extent to which the atmosphere attenuates radiation passing through it. It quantifies the reduction in the intensity of light as it travels through the atmosphere due to scattering and absorption by particles and gases.

The intensity is exponentially reduced in accordance with Beer-Lambert's law, which states that

$$\frac{I_i}{I_t} = e^{OD},$$

where I_i is the incident light's intensity, I_t is the transmitted light's intensity, and OD is the optical depth. The optical depth can be further broken down as

$$OD = -N \cdot \sigma_e \cdot L,$$

where N is the particle number density, σ_e is the extinction cross-section, and L is the length of the path travelled in the medium. The extinction cross-section, σ_e, is the sum of the absorption cross-section, σ_a, and the scattering cross-section, σ_s.

To gain an intuitive understanding of this expression it is useful to think of the mean free path length, $\overline{l_{fp}}$, which is the average distance between two interactions and corresponds to

$$\overline{l_{fp}} = \frac{1}{\mu_e} = \frac{1}{N \cdot \sigma_e},$$

where μ_e is the extinction coefficient. This means that OD is equal to the length of the medium divided by the average distance between two interactions so the optical depth corresponds to the average number of scattering/absorbing events along the distance L.

In simpler terms, a higher optical depth means that less light reaches the observer, indicating a denser or more absorptive medium. Optical depth is a key parameter in atmospheric sciences for understanding and modelling the effects of the atmosphere on the transmission of light, including phenomena such as visibility reduction, the greenhouse effect, and the behaviour of solar radiation.

2.3 LiDAR and Polarization

LiDAR, or Light Detection and Ranging, operates on principles similar to radar and sonar, emitting energy waves to detect and monitor objects. Unlike radar and sonar, LiDAR utilizes light, offering superior speed, precision, and resolution.

A LiDAR instrument sends out pulses of linearly polarized light. Some light is scattered back to the instrument, where it is measured. From this, the depolarization ratio, colour ratio, and total backscattering can be calculated. These are important optical properties used to characterize aerosols and clouds in the atmosphere.

Depolarization occurs when the polarization state of light changes upon interaction with a particle. This phenomenon is largely dependent on the shape, structure, and composition of the scattering particle. When light encounters a particle, the electromagnetic waves induce oscillations of electric charges within the particle. If the particle is symmetric and isotropic, such as a liquid droplet, these oscillations do not significantly alter the polarization state of the incident light, thus not causing depolarization. In contrast, an ice crystal, which is anisotropic and often has complex, non-spherical shapes, causes the scattered light to experience varying phase shifts and amplitude changes in different directions, resulting in depolarization. The difference lies in the uniformity of the internal structure and symmetry; liquid droplets are homogeneous and isotropic, whereas ice crystals have irregular, directional structures that disrupt the uniform scattering of light.

The depolarization ratio refers to the ratio of perpendicular to parallel backscattering. Aerosols and cloud particles tend to scatter light in a preferential direction, and the degree of depolarization can provide information about the shape and composition of the particles. For example, non-spherical aerosols and ice crystals in clouds exhibit higher depolarization ratios compared to more spherical particles like smoke. This difference in depolarization ratios is a useful parameter in atmospheric science for characterizing the physical properties of various atmospheric particles.

Colour ratio, also known as spectral colour ratio, is the ratio of the backscatter intensity at different wavelengths or colours of light. Different types of aerosols and clouds have unique spectral signatures due to their size, composition, and optical properties. By examining the colour ratio, researchers can distinguish between different aerosol types (e.g., dust, smoke, pollution). Additionally, by combining the information given by depolarization ratio and colour ratio researchers can also distinguish between cloud phases (e.g., water droplets, ice crystals). In general larger particles have a higher colour ratio. Ice particles are generally about $10\mu m$ which is much larger than smoke particles which are typically smaller than $1\mu m$. This results in the colour ratio of ice clouds being close to 1 while it is around 0.4 for smoke layers.

Total backscattering refers to the total amount of light scattered backwards by aerosols and clouds in the atmosphere. It provides information about the overall abundance and concentration of particles in the atmosphere. High values of total backscattering indicate the presence of dense aerosol layers or thick cloud formations, while low values suggest clearer atmospheric conditions.

Each of these optical properties offers valuable insights into the characteristics and behaviour of aerosols and clouds. Depolarization ratio helps discriminate between different particle shapes and compositions, colour ratio aids in identifying aerosol and cloud types based on their spectral signatures, and total backscattering provides a measure of the overall particle abundance in the atmosphere. Together, these parameters contribute to a better understanding of atmospheric processes, climate dynamics, and environmental impacts associated with aerosols and clouds.

In this project, the primary objective was to distinguish ice clouds from other atmospheric structures. The easiest way to do it qualitatively is by examining the depolarization ratio. Ice clouds have significantly higher depolarization ratios, ranging from 0.3 to 0.5, compared to values around 0.15 for smoke layers. Hence, it follows that the depolarization ratio would be the best input for an AI as well when differentiating. This reasoning, coupled with computational constraints, led to our decision to solely focus on the depolarization ratio. Introducing even just one additional physical input would have doubled the amount of data and memory required during training, a trade-off that was deemed unjustifiable.

2.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep neural networks that have gained significant popularity, particularly in computer vision tasks such as image classification and object detection. Unlike traditional neural networks, which process data such as images as a flat vector, CNNs are designed to efficiently handle grid-like data, such as images, by preserving the spatial structure. CNNs are characterized by their ability to automatically learn hierarchical patterns and features directly from raw data. They achieve this through the use of convolutional layers, which apply filters to the input data to extract features, and pooling layers, which downsample the feature maps to reduce computational complexity and extract the most important information. One of the key advantages of CNNs over traditional neural networks is their ability to exploit spatial locality by using local receptive fields, where each neuron in a convolutional layer is connected to a small region of the input image. This unique architecture, which is illustrated in Figure 3, enables CNNs to capture local patterns and relationships, making them ideal for tasks involving spatial data.

In machine learning, there are numerous parameters that shape the structure of a model, but equally vital are the hyperparameters that dictate the model's training process. Among these hyperparameters, the choice of loss function stands out as a critical metric in evaluating a model's performance during training. Analogously, one can

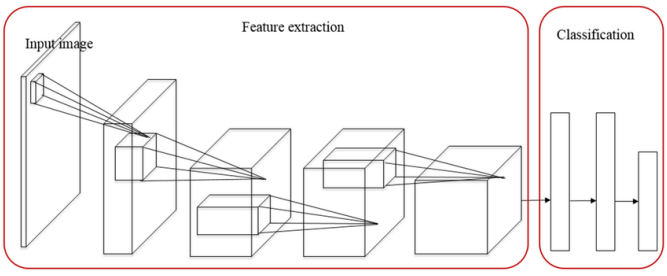

Figure 3: The structure of a CNN taken from a paper by Lin et al.[4]

envision the training process as
navigating a landscape to reach the lowest point, akin to finding the global minimum of the loss function. Each iteration of training corresponds to the model making incremental adjustments, akin to a needle traversing the uneven surface, propelled by the gradient of the loss function.

In supervised learning, where the model is trained on labelled data, the loss function typically quantifies the disparity between predicted and true values, guiding the model toward convergence. However, in the realm of unsupervised learning, where labelled data is absent, determining the efficacy of the loss function becomes more challenging. Without ground truth labels to guide the optimization process, discerning whether the loss function's minima aligns with the desired solution or holds any physical significance becomes uncertain.

To address this uncertainty, a common strategy involves initially training the model on a similar dataset with labelled examples, allowing for the evaluation of the architecture and loss function in a supervised context. Once validated, the learned architecture and loss function can then be transferred to the unsupervised problem domain. In this scenario, only the hyperparameters require qualitative evaluation, typically through visual inspection of the final output.

This approach mitigates the ambiguity surrounding the loss function's efficacy in unsupervised settings, leveraging insights gained from supervised learning to inform the design and evaluation of models in unsupervised contexts.

In this project, various prebuilt methods were used from "scikit-learn" and "PyTorch". Additionally, inspiration was taken from the code of various papers. Most notably "Unsupervised Learning of Image Segmentation Based on Differentiable Feature Clustering" by Kanezaki et. al [5] from which the CNN's structure and loss function were copied. In this way, we are ensured of their efficacy and need only to evaluate the hyperparameters and the later parts of the segmentation process.

2.5 Semantic Segmentation

In this project, the objective is to discern various structures within images, a task achieved through semantic segmentation. Semantic segmentation involves assigning every pixel in an image with a corresponding label or category, enabling the identification of distinct objects or regions within the image. This technique is applied in various computer vision tasks, notably in training self-driving cars to recognize and navigate through different elements in their surroundings. Despite its efficacy, traditional semantic segmentation methods are not particularly prevalent due to their reliance on large, meticulously labelled datasets. Creating such datasets is a laborious and resource-intensive process, requiring extensive manual annotation of pixel labels, making it impractical for many applications. For example, in this project, a single image contains over 30 million pixels before downscaling. To circumvent this limitation, researchers have explored unsupervised approaches, which introduce a new set of challenges and considerations.

In unsupervised learning, models identify patterns and structures within data without

guidance from labelled examples. However, it is very difficult to assess the validity of these structures. Evaluating an unsupervised method is always a challenge and the best method is usually to qualitatively assess the quality of the final results.

In Figure 4, we observe an image sourced from the Cityscapes dataset[6]. This dataset employs colour-coded annotations to represent various objects within the scene; for instance, cars are depicted in blue, while houses appear in grey. During supervised semantic segmentation, these colour labels serve as targets for training algorithms. However, in unsupervised learning scenarios, such labels are absent, necessitating alternative methods for segmentation without explicit guidance.

Figure 4: Example image from the dataset Cityscapes[6]. On the right side, colours corresponding to different classes can be seen for each pixel.

Semantic segmentation relies on CNNs to segment images into semantically meaningful regions. Understanding the basic building blocks of CNNs is paramount for comprehending their role in semantic segmentation tasks. At the core of CNN architecture lie convolutional layers, which extract intricate spatial features from input images using learnable filters. These filters convolve across the image, generating feature maps that capture hierarchical representations of the input. Pooling layers, another essential component, aid in spatial dimension reduction by downsampling feature maps, facilitating computational efficiency, and preserving important spatial information. Activation functions, such as the Rectified Linear Unit (ReLU), introduce non-linearities to the feature maps, enabling the network to learn complex relationships between features. Stacking these layers together forms a robust framework for semantic segmentation, allowing CNNs to discern object boundaries and classify pixels into distinct semantic categories.

In semantic segmentation tasks where pixel-level accuracy is crucial, pooling layers may discard fine-grained spatial details, making it challenging for the model to precisely delineate object boundaries. It is probably for this reason that they were omitted from the CNN structure outlined in the paper by Kanezaki et. al[5].

2.6 Dimensionality Reducing Techniques

The dimensionality of the features extracted by the CNN is decided by the number of channels in the network. The suitable number of channels is decided by the complexity of the task, the size and nature of the input data, and the architecture of the network. For complex tasks like semantic segmentation, where the network needs to capture detailed spatial information, CNNs may have a large number of channels, often in the range of several thousand to tens of thousands. However, the more channels the more memory is required during the training process. As a consequence, the resulting feature space

consists of 64 dimensions, which qualifies as high-dimensional data. This can potentially introduce challenges during the classification process. Therefore, it can be beneficial to first employ a dimensionality-reducing technique such as Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE).

Principal Component Analysis (PCA) is a linear dimensionality reduction method commonly used for feature extraction and data visualization. It transforms high-dimensional data into a lower-dimensional space by identifying orthogonal vectors, called principal components, that capture the directions of maximum variance. By selecting a subset of these principal components, PCA effectively reduces the dimensionality of the dataset while preserving as much variance as possible.

On the other hand, t-Distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear dimensionality reduction technique that aims to preserve the local structure of the data. It emphasizes relationships between nearby data points in the high-dimensional space, by modelling pairwise similarities using t-distributions.

In terms of computation, PCA is more efficient and scalable to large datasets compared to t-SNE, which can be computationally intensive, especially for high-dimensional data. Additionally, PCA provides interpretable components representing directions of maximum variance, making it suitable for understanding the global structure of the data. Conversely, t-SNE is primarily used for visualization purposes and lacks a straightforward interpretation of its low-dimensional embeddings.

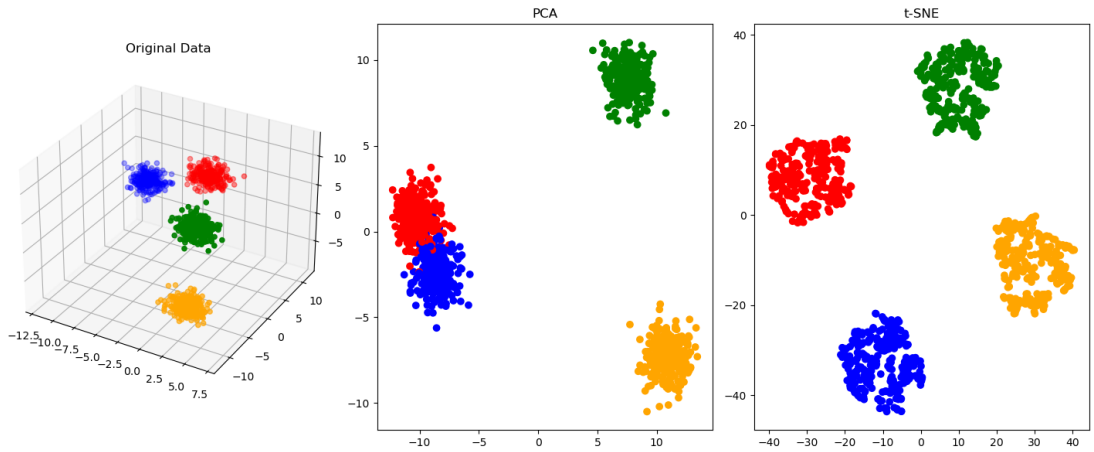

Figure 5: Comparison of the two dimensionality reducing methods: PCA and t-SNE.

In Figure 5 we can see both PCA and t-SNE being applied to example data generated using scikit-learn's function "make blobs". In this case, we can see that PCA prioritizing the global structure makes two of the blobs be pushed together while t-SNE keeps them all separate. Conversely, t-SNE fails to capture the scale of the image changing both axes from ± 10 to ± 40.

In summary, PCA is suitable for linear dimensionality reduction and capturing global structure, while t-SNE is ideal for visualizing high-dimensional data and preserving

local structure. The choice between PCA and t-SNE depends on the specific goals of the analysis and the characteristics of the dataset.

2.7 Classification Methods

Classification methods are ways in which we separate data into different classes. Countless such methods can be employed but in this project, we focused on three: KMeans clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and argmax.

KMeans and DBSCAN are two popular clustering algorithms used in machine learning and data analysis. KMeans is a partition-based clustering algorithm that aims to partition a dataset into a predefined number of clusters, where each cluster is represented by its centroid. The algorithm iteratively assigns each data point to the nearest centroid and then updates the centroids based on the mean of the data points assigned to each cluster. KMeans is simple to implement and computationally efficient, making it suitable for large datasets.

On the other hand, DBSCAN is a density-based clustering algorithm that groups closely packed points based on a density criterion. DBSCAN does not require the number of clusters to be specified in advance and can automatically detect clusters of arbitrary shapes and sizes. It works by defining two parameters: epsilon (eps), which specifies the radius within which points are considered neighbours, and MinPts, which specifies the minimum number of points required to form a dense region. DBSCAN identifies core points, which have a sufficient number of neighbours within the epsilon radius, and expands clusters by recursively adding neighbouring points.

The application of both KMeans and DBSCAN can be seen below in Figure 6 on the same data as in Figure 5.

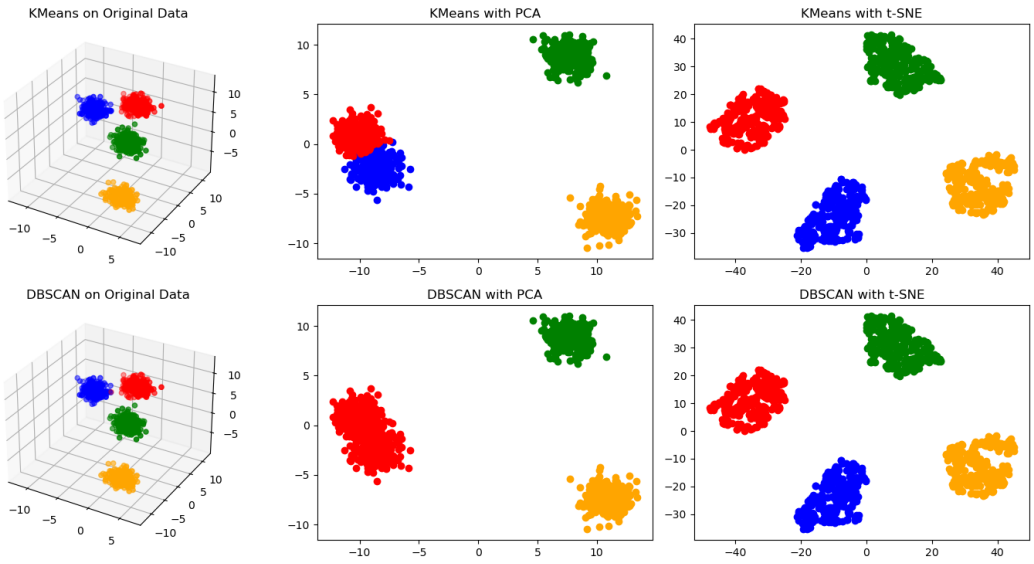

Figure 6: Comparison of the two clustering methods KMeans and DBSCANS

In this example, it is clear that DBSCAN when combined with PCA performs the worst, but this does not mean that this combination will always underperform or that DBSCAN is worse than KMeans. It would have been just as easy to construct an example in which KMeans failed. Ultimately, the best choice of classification method depends on the data. While KMeans is suitable for datasets with well-defined clusters of a known number, DBSCAN is more robust to noise and capable of discovering clusters of varying shapes and densities. However, DBSCAN may struggle with datasets of varying densities or clusters with irregular shapes. Both algorithms have their strengths and weaknesses, and the choice between them depends on the specific characteristics of the dataset and the desired outcome of the clustering task.

This dependence highlights the true power of PCA and t-SNE. Reducing the dimensions to two or three allows us to visualize the data and from it determine the proper choice of classification method.

Finally, if neither KMeans nor DBSCAN yields satisfactory results due to the complexity of the structures, it may be worthwhile to employ argmax classification. This is the approach that Kanezaki[5] used while employing the same CNN structure and loss function. Unlike KMeans or DBSCAN, argmax classification does not rely on clustering; instead, it assigns each data point to a class based on which dimension it exhibits the highest value. To function properly, normalization of the different dimensions is essential, along with a large number of dimensions.

3 Method

The cornerstone of our approach lies in the training of a CNN. However, CNNs can be very sensitive to noise. This is because they are designed to exploit spatial locality by using local receptive fields, which means that each neuron in a convolutional layer is connected to a small region of the input image. If noise is present in these local regions, it will affect the network's ability to extract meaningful features. Hence, meticulous data preparation or preprocessing is imperative.

Furthermore, the data also had to be downscaled before being fed to the network. This is because the training process is memory-intensive and contrary to other types of computational challenges this did not simply make the program slow but made it crash. Thereby, imposing an absolute limit on the amount of memory that could be used. The depolarisation data for the period investigated amounts to about 250 GB. This made downscaling an obvious step when trying to find the best way to use the limited memory. Next, the data was further preprocessed to remove as much noise as possible while preserving information and a high level of detail.

The CNN was trained on a dataset encompassing a diverse array of aerosol types, atmospheric conditions, and geographical locations. This diverse dataset aimed to equip the model with the capability to generalize effectively across varied scenarios.

After the training of CNNs, the images were fed to them to create feature maps. The maps were then used for classification using various methods. Some of these methods

needed dimensionality reduction to perform optimally so various such techniques were tested as well.

The various parts of the process could not be evaluated in isolation. Therefore, they all needed to be evaluated in combination with each other. This was necessary to determine the best approach.

3.1 Data Considerations

A critical factor for any deep learning success is the quality and scale of the training dataset. The size of the dataset is crucial, due to the delicate balance between representativeness and computational efficiency. While a larger dataset enhances the model's ability to generalize across diverse scenarios, it concurrently increases the computational intensity of the training process.

Other important aspects are the data's amount of noise and high resolution. The presence of noise often confuses image classification models, likely necessitating averaging methods. The high resolution, while advantageous for detailed analysis, poses computational challenges due to increased processing demands.

Downscaling and smoothing out the images need to strike a balance between retaining essential information and reducing computational complexity. This preprocessing step aims to optimize the dataset for effective model training, ensuring that the convolutional neural network captures essential patterns without succumbing to the computational burden posed by excessive data size and resolution.

The data provided by CALIPSO has a very high level of detail but this is accompanied by significant levels of noise. Due to the sheer volume of data available, constraints are necessary. We will limit ourselves to night data from the period of December 2019 to March 2020. This still amounts to about 800 gigabytes of data which already is quite demanding for a normal laptop. Night data was chosen because it has significantly less noise and that period was chosen because there were several large forest fires during it.

3.2 Preprocessing

For any task involving artificial intelligence (AI), the preprocessing step is crucial for achieving success. This process entails striking a delicate balance between reducing noise and preserving vital information. In Figure 7, we observe depolarization data from a CALIOP scan conducted on January 4, 2020. At this point, the only preprocessing steps performed were removing data points above 35 km and below 3 km relative to the tropopause by setting their values to zero. However, in subsequent stages, additional noise reduction will occur by assigning their values to 0.01. The colour map employed represents values at or below zero as black, thus any eliminated noise will blend in with the sky background, depicted in blue.

Including data from up to 3 km below the tropopause enhances model training by capturing a more extensive range of ice clouds, which predominantly lie beneath this bound-

ary. This broader dataset aids in better distinguishing between various features within CALIOP data, ultimately improving classification accuracy.

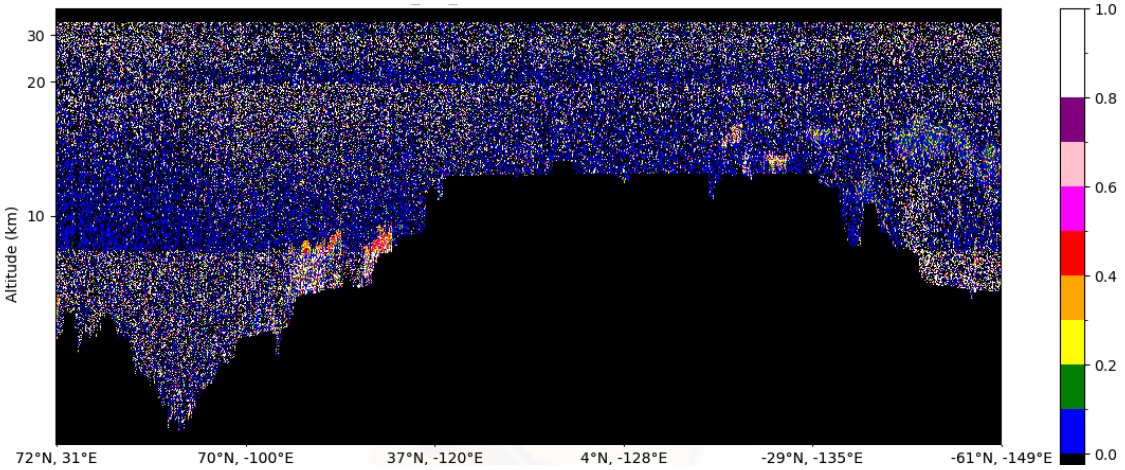

Figure 7: Depolarization ratio of data from the fourth of January 2020. All data above 35km or below 3km below the tropopause has been removed.

The decision to discard data above 35 km stemmed from its excessive noise levels, rendering it unsuitable for analysis. Similarly, data below 3 km below the tropopause was omitted as our focus lay solely on the stratosphere. Given the variable nature of the tropopause's altitude, a fixed height threshold couldn't be universally applied. Its altitude varies significantly depending on local convection activity, resulting in higher tropopause levels in warmer regions, such as near the equator, and lower levels in colder regions, such as near the poles.

The initial step in mitigating noise involves filtering out laser pulses with unusually high root mean square (RMS) noise levels. To determine the threshold for such noise levels, an orbit of noise data was plotted, as depicted in Figure 8. Analysis of the plots revealed that while some values were exceptionally high, the majority clustered into two distinct bands, corresponding to each channel. Based on this observation, a decision was made to remove laser pulses where the RMS noise exceeded 85 for the perpendicular channel or 70 for the parallel channel. In the specific orbit analyzed, this action resulted in the removal of 11% of the pulses, and it was about the same for the rest of the orbits within the period studied.

While this is a significant portion of the data, it's crucial to recognize that this issue is highly localized. The South Atlantic Anomaly (SAA), located near South America, is a region where the Earth's Van Allen radiation belt is notably weaker than normal. As a result, a higher influx of charged particles from the sun penetrates deeper into the atmosphere, leading to interference with satellite signals. The majority of the affected data will be concentrated in this region, meaning that data from all other regions will be significantly less affected.

The RMS noise is computed onboard for each laser pulse by evaluating the standard deviation of 1000 samples acquired within the 65-80 km altitude range, each spanning

15 meters. Lidar measurements are subject to two main sources of random error: variations in the received laser scattering signal from the atmosphere and fluctuations in the background signal. Both components are essential considerations in estimating the overall random error. The random error attributed to the scattering signal can be approximated using a noise scale factor. The theoretical basis for calculating this factor is rooted in the understanding that photons from solar background radiation adhere to a Poisson stochastic process[7].

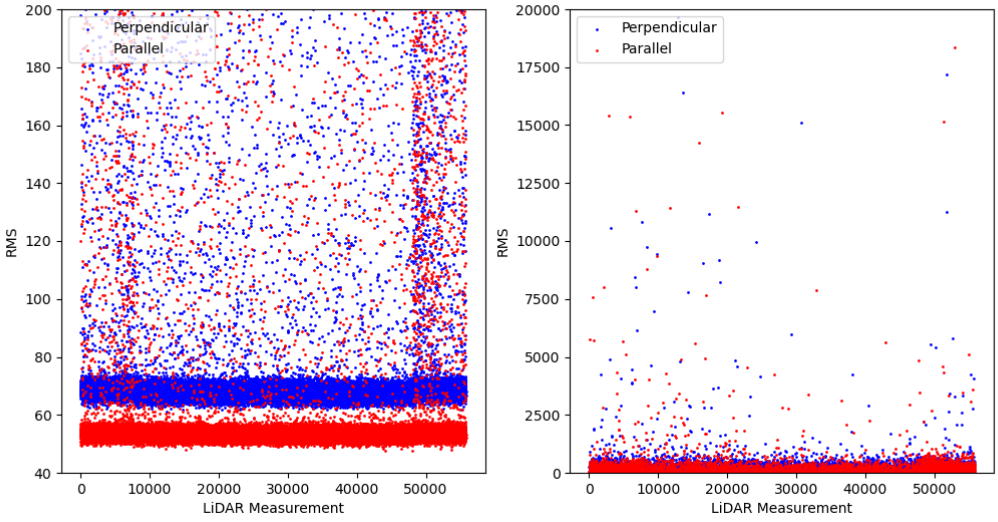

Figure 8: Perpendicular and parallel RMS noise for the same orbit as the data in Figure 7. The difference between the subplots is solely the scale on the y-axis.

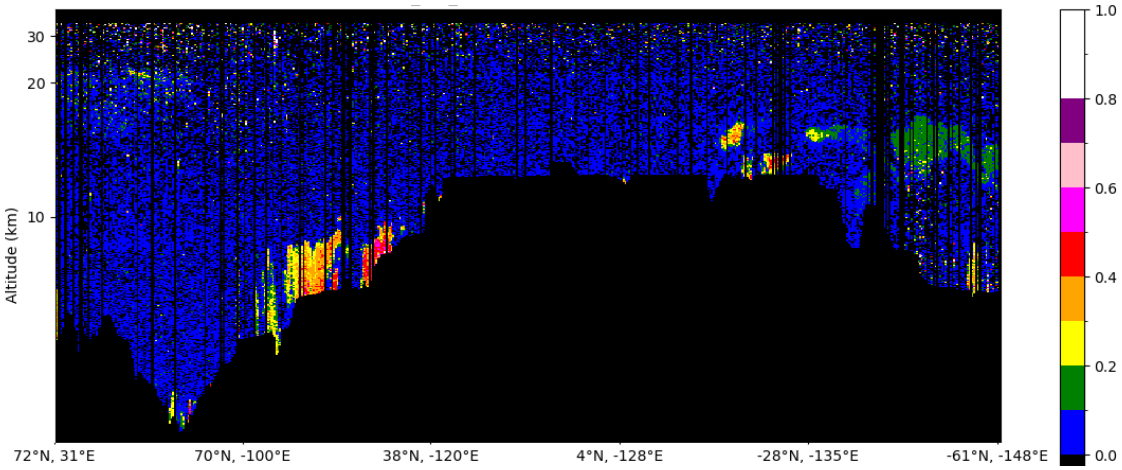

Figure 9: The same raw data as in Figure 7. All data points where the RMS of parallel or perpendicular backscattering exceeded 70 or 85, respectively, were excluded. Additionally, it has been down-scaled horizontally by a factor of 100

Moreover, given the computational constraints of the project, downsampling was necessary. We opted to downscale all data horizontally by a factor of 100. This transformation reduced the length of the horizontal dimension from over 55 thousand to 556, effectively

decreasing the horizontal resolution from approximately 300 meters to 30 kilometres. The impact of these two procedures on the same orbit as previously analyzed is illustrated in Figure 9 above.

At this point, the data has been processed enough to be used for training a CNN. In fact, these models perform quite well, which is reasonable since the relevant features can now be easily identified qualitatively. Both the big orange-red structures to the left and the smaller structures with the same colour to the left are ice clouds. The green patch that can be seen to the left is a large smoke layer.

However, the models trained on this data do still struggle with the noise still present, especially in the top few kilometres of the data. To further reduce noise the next step taken was to set all values below 0 or above 1 to 0.01. This can be done because such values are not physical. Next, the data removed due to having too high RMS, which we can see as vertical black lines in Figure 9, were filled in using linear interpolation. Lastly, connected component analysis (CCA) was performed to remove noise. The data in the image was converted to binary using 0.05 as a threshold. Then CCA found all connected structures of 1s which allowed all below a certain size to be removed.

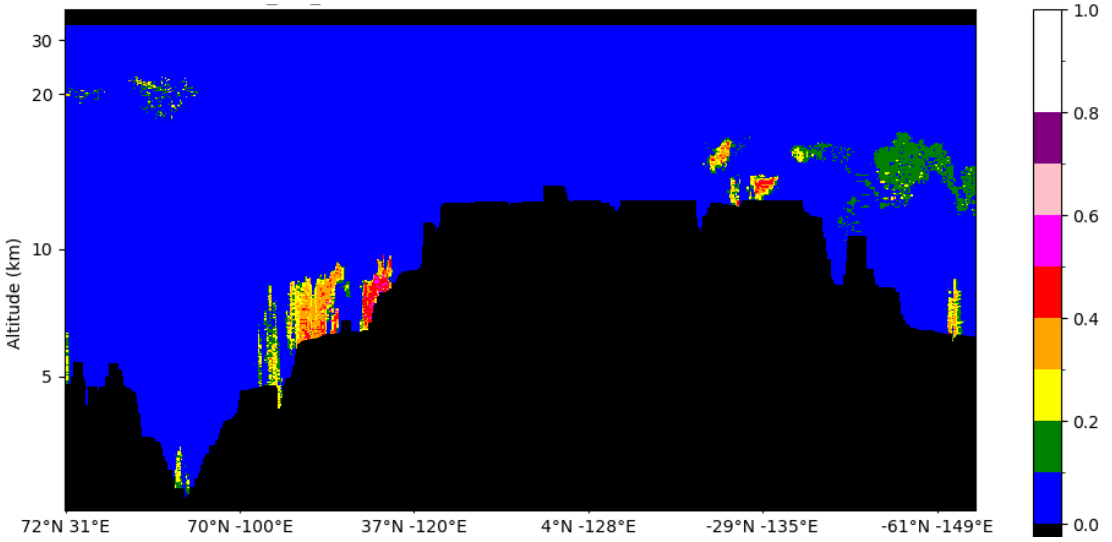

Figure 10: The same raw data as in Figure 7 and Figure 9. Further, preprocessing has been done consisting of setting values above 1 and below 0 to 0.01, filling in the missing lines using interpolating, and removing noise using CCA.

The problem with removing noise using CCA is that it fails to do anything to the noise inside of larger structures. For example, in the smoke layer on the right side in Figure 10 we can see a bunch of noise that the CCA did not remove because it was not isolated. One common technique to deal with such noise is to blur the image using convolutional filtering. The disadvantage of such techniques is that at the edges of the structures they will be averaged with the background creating outlines of about half the ratio the structure exhibits. Such outlines are not physical but would be present in every image. We tried blurring the images this way and trained several of the CNN on them but the results were highly lacklustre.

4 Results & Analysis

The process of optimizing hyperparameters was very similar for all of them. Therefore, We will only show it for a few: continuity step size, learning rate, and the choice of optimizer. Let us start with continuity step size.

Both the continuity step size ($step_{con}$) and the similarity step size ($step_{sim}$) are parts of the loss function. In this context, $step_{con}$ determines the spacing between adjacent sampling points along a feature's orientation, while $step_{sim}$ controls the spacing between features or descriptors in the image space. If $step_{con}$ is much larger than the $step_{sim}$, the emphasis shifts towards producing a smoother output, prioritizing global feature consistency over local detail preservation. Conversely, if the $step_{con}$ is much smaller than the $step_{sim}$, the focus is on capturing fine-grained details, potentially at the expense of computational efficiency. Achieving an appropriate balance between these sizes is critical for optimizing the trade-off between output smoothness and detail preservation.

We found that an absolute size of 3 was appropriate for the $step_{sim}$ and that $step_{con}$ should be above 10. Next, we tried three different values for $step_{con}$: 15, 20, and 25. We looked at images for around fifty orbits for each model to ensure that the evaluation was right for the entire data set. One such comparison can be seen below in Figure 11. This orbit is the same as was illustrated in the section on preprocessing. Throughout the comparisons, we will stick to visualize this orbit for the convenience of the reader.

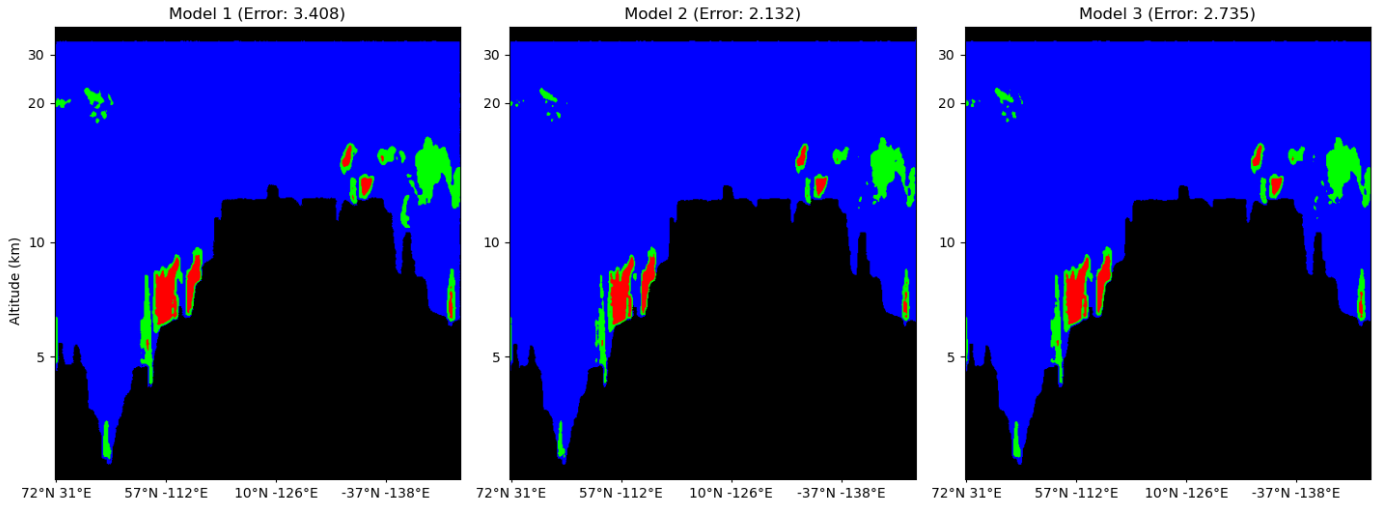

Figure 11: Segmented images resulting from Kmeans Clustering being applied to the features of three different but similar CNN models. The difference between them is the size of the hyperparameter $step_{con}$ which is the step size for computing continuity loss.

From Figure 11 it is hard to tell which model performs better and this was quite typical for the process. Optimizing hyperparameters qualitatively is difficult partially because the results often look very similar. In this case, we determined model 2 to have performed the best largely because it had the lowest error. This error comes from the KMeans clustering which was performed on the feature map after reducing its dimensions from 64 to 2 using PCA.

At this point, we were not using any dimensionality-reducing techniques and we only tried KMeans clustering as a form of classification. Next, we will show the process by which we evaluated different types of dimensionality-reducing and classifying methods. Let us start by again using PCA and KMeans but on the three models that we will use for the rest of the comparisons.

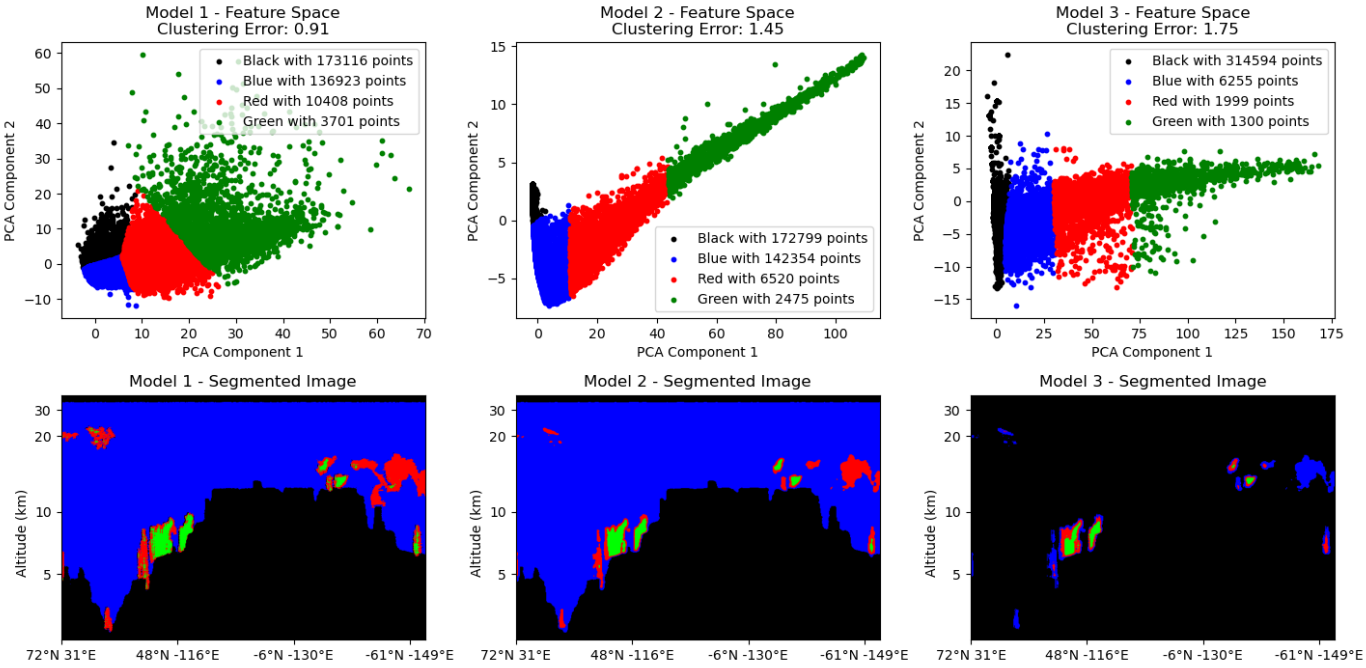

Figure 12: Comparison of three different models. The top row shows the final output in feature space after PCA has been used to reduce the dimensionality from 64 to 2. The bottom row shows the corresponding segmented images. The colours are based on the KMeans clustering of the data in the top images.

In Figure 12 we can see the feature space of three different models each reduced to two dimensions using PCA. Model 1 was trained on less processed data such as that which can be seen in Figure 9. Conversely, both models 2 and 3 were trained on the more processed data such as that which can be seen in 10. What separates models 2 and 3 is that model 3 was trained for 10 iterations instead of 5 and with a learning rate of 0.1% instead of 0.01%. Furthermore, model 1 was trained using a simpler loss function, smaller batch size, and half as many convolutional layers. These differences result in very different shapes in the feature space. However, the resulting segmented images are most similar for models 1 and 2 with model 3's being the odd one out. This suggests that models can converge on the same answer while using vastly different approaches.

Something important to understand is that the clustering error of the KMeans method does not necessarily correspond to the better result when comparing different methods. As can be seen in Figure 12 the clustering error is lowest for model 1 at 0.91. However, the resulting image is quite pixelated with tiny green dots appearing within otherwise red structures. Such artefacts are undesirable, particularly given the typically smooth nature of physical structures at this scale. Therefore, we would assert that Model 2

achieved the best performance among these three models.

Next, let us investigate what happens if we use t-SNE instead of PCA to reduce the dimensionality. We compared the same three models from before, reducing their dimensions from 64 to two and then performing KMeans clustering. The result can be seen below in Figure 13.

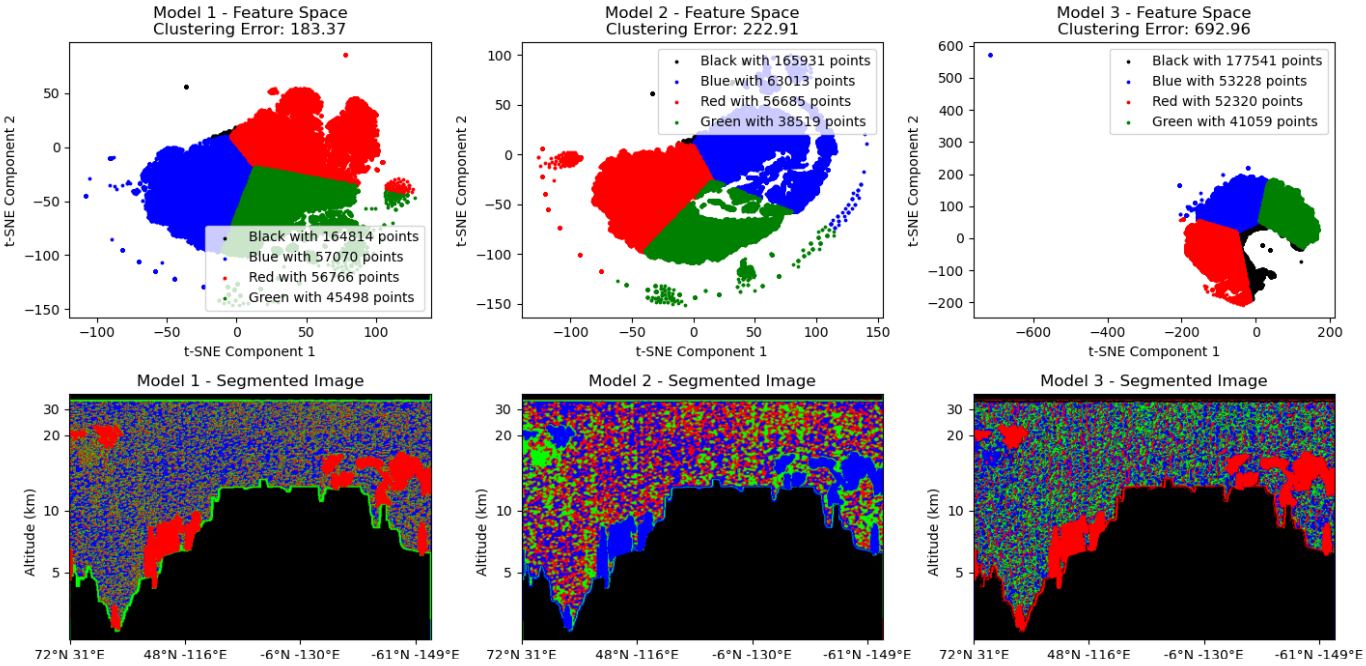

Figure 13: Comparison of three different models. The top row shows the final output in feature space after t-SNE has been used to reduce the dimensionality from 64 to 2. The bottom row shows the corresponding segmented images. The colours are based on the KMeans clustering of the data in the top images.

From the top row of Figure 13, we can see that the shapes of the data are quite different for each model after t-SNE as compared to after PCA. From the bottom row, we can see that these new shapes are not well-suited for KMeans. In each segmented image, the important cloud and aerosol structures are present but the rest of them are complete nonsense.

A possible explanation for this failure could lie in the choice of hyperparameters. For this visualization, we used the standard values in "sci-kit learn" except for perplexity which we set to 50. According to the original paper for the method, "*The performance of SNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50.*"[8] However, this might not necessarily be the case when dealing with large datasets. In t-SNE, all points repel each other but only a few attract each other. Perplexity is the number of neighbours that each point is attracted to. This means that using a perplexity of 50 when there are over 300 thousand points might make the method struggle to find any larger structures.

We tried increasing the perplexity but the largest our computer could manage without

crashing was 300 which only produced marginally better results. It is possible that the result could improve drastically for a much higher perplexity but for now, it seems that t-SNE is not as well suited for this task as PCA. At least not using KMeans clustering. We will also investigate the classification methods DBSCAN and argmax. Argmax will be performed without any dimensionality reduction but DBSCAN will also be investigated using both PCA and t-SNE.

Let us start by trying DBSCAN with PCA. The result of this can be seen below in Figure 14.

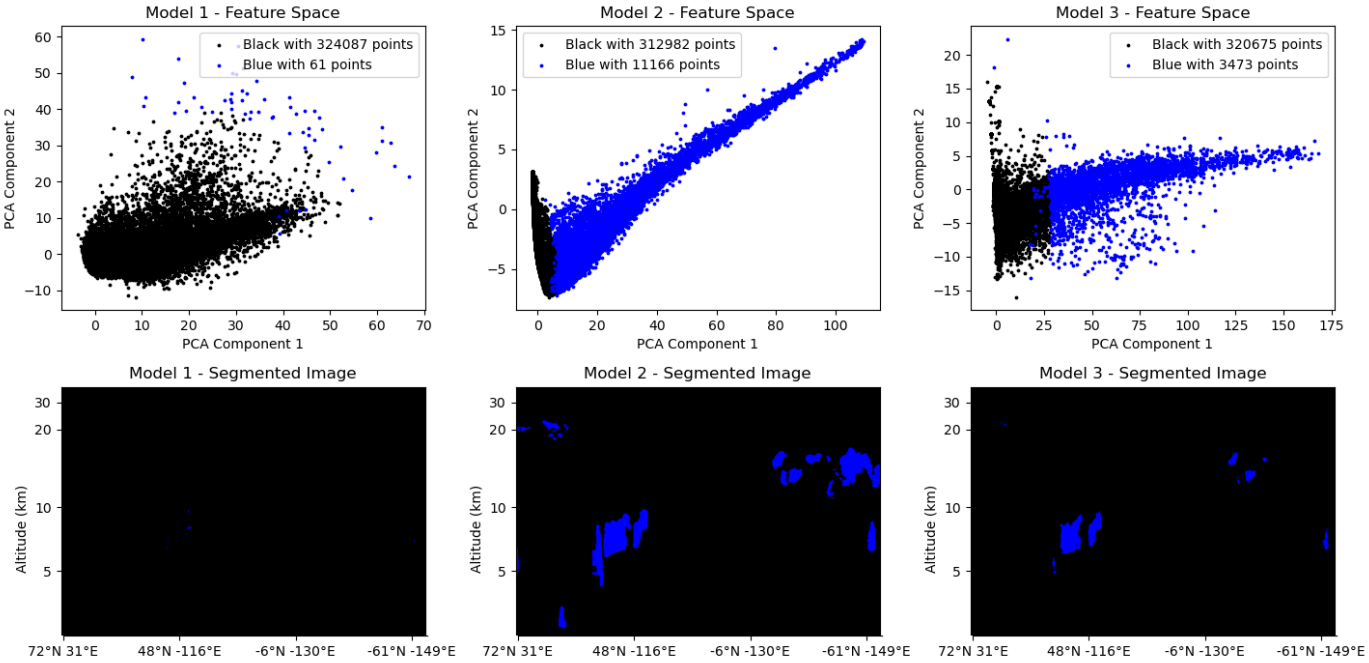

Figure 14: Comparison of three different models. The top row shows the final output in feature space after PCA has been used to reduce the dimensionality from 64 to 2. The bottom row shows the corresponding segmented images. The colours are based on DBSCAN clustering of the data in the top images.

In Figure 14 we can see that this combination does not perform particularly well. The segmented images are almost entirely black with some small parts of the important structures showing as blue. The result is underwhelming, but there is a reason for this. DBSCAN is a method that scales by the number of points squared making it very computationally intensive for large datasets. This limitation forced me to perform the algorithms in batches which is quite problematic for a clustering method. The largest batch size our computer could handle was 40 thousand which is about 12% of the total data points for each image. It should be noted that it is important to select these points randomly otherwise the results will be complete nonsense.

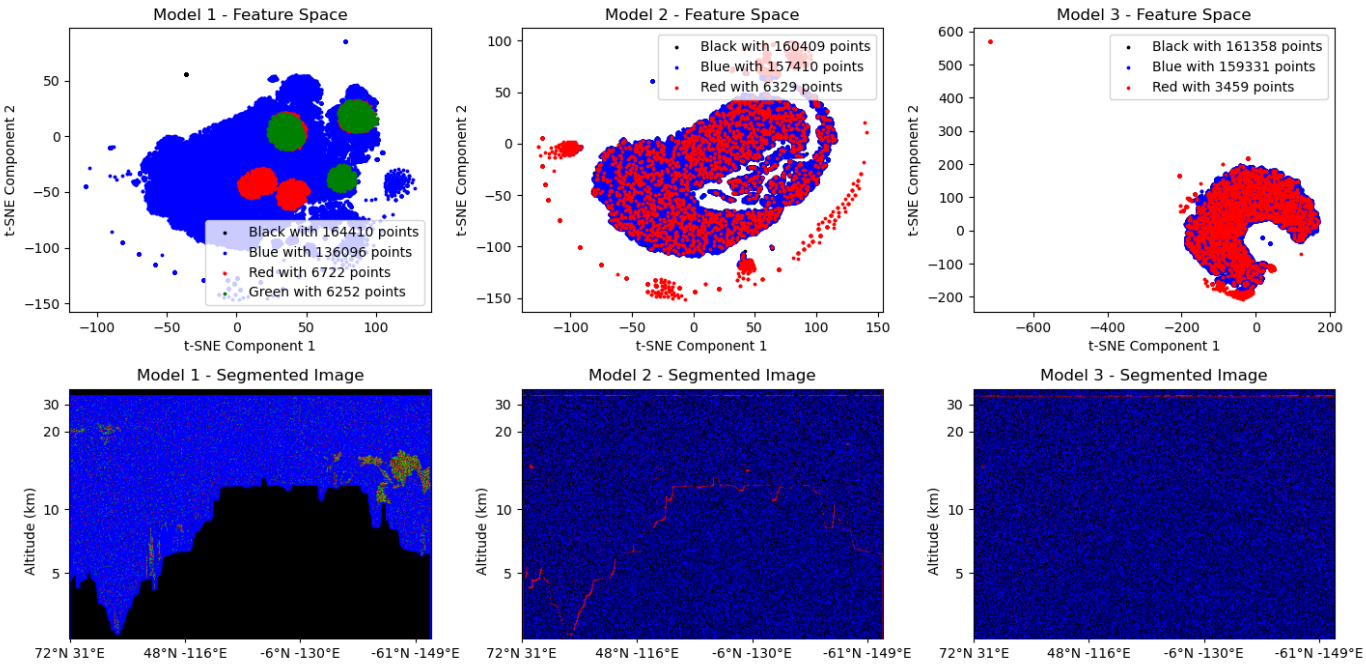

Figure 15: Comparison of three different models. The top row shows the final output in feature space after t-SNE has been used to reduce the dimensionality from 64 to 2. The bottom row shows the corresponding segmented images. The colours are based on DBSCAN clustering of the data in the top images.

Next, let us see if DBSCAN performs better when paired with t-SNE. As can be seen in Figure 15, while DBSCAN does perform differently when paired with t-SNE it can not be said to be much better. While the resulting segmented image for model 1 is probably the best of any involving DBSCAN, there is a significant amount of noise in the output. Moreover, the images from models 2 and 3 are terrible, making the method highly unreliable. We believe this not to be a failure of the method itself but rather of the fact that the clustering had to be performed in batches.

Something to note is that the performance of DBSCAN depends heavily on the size of the hyperparameter "epsilon". The best size of this parameter can be estimated by finding the elbow point in a k-distance graph. This by itself is quite time-consuming and needs to be repeated for each image and model.

The lacklustre performance of DBSCAN is quite disappointing because it has a very attractive feature that KMeans lacks. DBSCAN is not forced to have the same amount of clusters in each image. This means that if a certain type of cloud is not present it will not try to look for it. We can see this in Figure 15 where it only finds 2 clusters for model 3 but 3 clusters for the other two models. This does not help its performance in this particular case, but in theory, it is a very powerful feature.

The last classification method to evaluate is argmax. This method requires a large number of dimensions so it will not be paired with either PCA or t-SNE. The result of it being applied to the same orbit and models as the previous comparisons can be seen below in Figure 16.

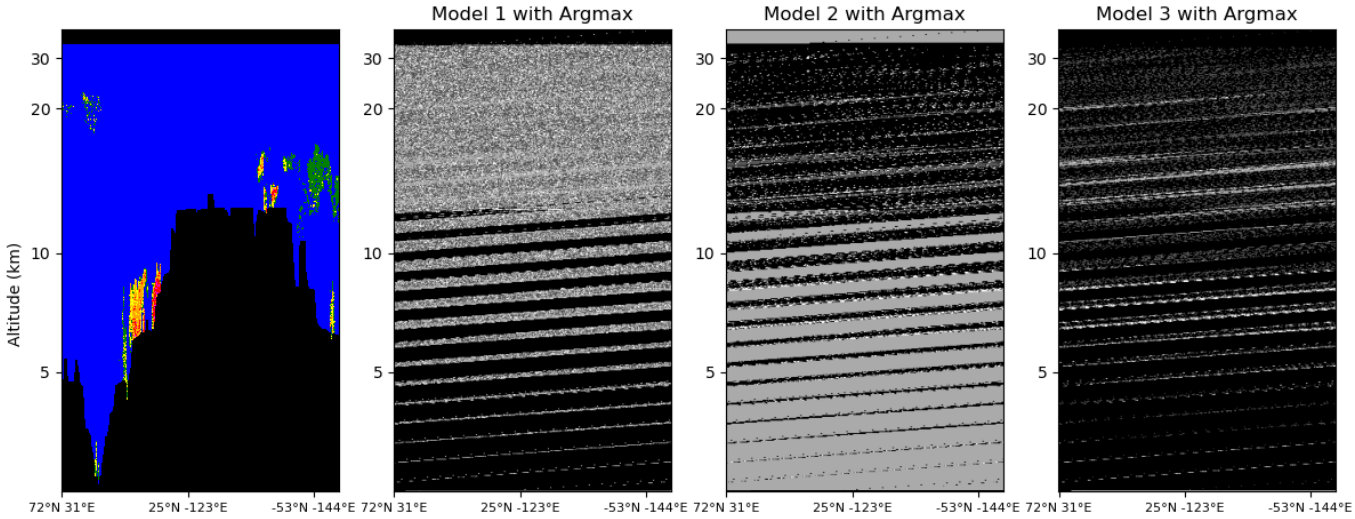

Figure 16: From the left: The first plot is the data fed to each model, and the next three are the results of using Argmax classification.

As can be seen in Figure 15, argmax fails completely to find anything of value. This could be because the number of channels is too small but we doubt that this is the sole explanation. When Kanezaki[5] employed this classification method they then incorporated it as a part of the training process. They then used the resulting labels as pseudo-targets. This means that during the training process, the argmax classification will gradually improve along with the training. This facilitates the approach and is probably necessary for its viability.

Some other aspects of the method that needed to be tinkered with were the number of epochs and the choices of learning rate and optimizer. Learning rates of both 0.1% and 0.01% were tested for the optimizers Stochastic Gradient Descent (SGD) and Adam.

SGD works by updating model parameters based on the gradients of the loss function with respect to the parameters. It does this iteratively using small batches of training data. The update rule for SGD is straightforward: it subtracts a fraction of the gradient from the current parameter values, scaled by a learning rate.

Adam, on the other hand, is a more advanced algorithm that adapts the learning rate for each parameter. It maintains moving averages of both the gradient and the squared gradient. These moving averages are then used to compute adaptive learning rates for each parameter. The update rule for Adam involves several steps, including computing these moving averages and adjusting the parameters accordingly.

In terms of comparison, Adam often converges faster than SGD, especially in scenarios with noisy or sparse gradients, thanks to its adaptive learning rate mechanism. However, Adam introduces additional hyperparameters and requires more memory and computational resources compared to SGD. Despite this, SGD can be more robust to noisy data due to its simplicity, while Adam may be more sensitive to noisy gradients.

In Figure 17 we can see a comparison of four different loss curves with crosses to indicate validation loss at that point. From these, it seems clear that Adam performs better overall, even though the validation loss is somewhat unstable.

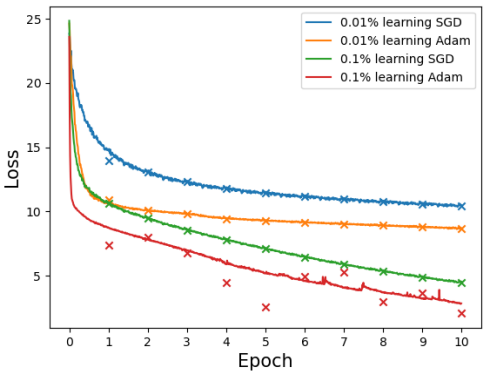

Figure 17: Comparison of loss curves for four different models. Two of these used SGD as their optimizer while the other two used Adam. One of each pair used a learning rate of 0.1% while the others used 0.01%. The curves are training loss while the crosses are validation loss.

Normally when training a CNN it is enough to try and minimize the loss for the validation set. However, in this case, we found that this did not necessarily lead to a good final result. Models that were trained for more than 5 epochs tended to become overly fixated on noise and miss the larger structures. This is problematic and suggests that either the loss function or the classification method is not ideal.

To further investigate this as well as the proper amount of clusters for KMeans, three otherwise identical models are compared in Figure 18. $Model_1$, $Model_5$, and $Model_{10}$ were trained for one, five, and ten epochs respectively. From the left side of the figure, we can tell that all three models perform quite similarly on average. From the right side image of the figure, we can see that $Model_1$ performs quite differently from the other two. It has many more high peaks but when it does not have a peak it generally has the lowest error.

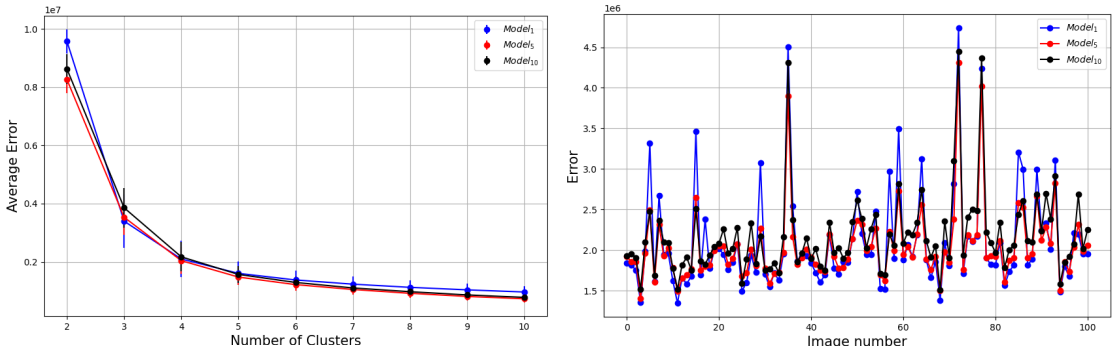

Figure 18: Comparison of three CNN models clustering performance, each trained with a batch size of 16, 64 convolutional layers, a similarity step size of $step_{\text{sim}} = 3$, and a continuity step size of $step_{\text{con}} = 20$. The models were trained for one, five, and ten epochs, respectively. The right image displays the error for the first 100 images in the dataset using four clusters. The left image presents the average error for these 100 images across 2-7 clusters.

For $Model_5$ and $Model_{10}$ the error is quite constant except at three points. For images 35, 72 and 77 there are clear peaks for all models. This suggests that there is something in these images which the models struggle to cluster effectively. To investigate further these images were plotted and it was found that for these images all three models failed to differentiate between the background and the removed regions (blue and black in Figure 10).

For comparison, we have included classification performed using the SIBYL algorithm[9] with Figures 19 and 20 in the appendix. This is for the same orbit as has been illustrated continuously throughout the paper but it has been scaled differently and cut up into pieces. This makes it quite hard to compare its features to the ones in our models. There are several advantages of this algorithm compared to the ones outlined in this paper. It can find many more different kinds of structures and has a very high degree of detail. However, there are also disadvantages. It has a strong tendency to divide up structures into many thin stripes alternating between two or more classes. For example, in the bottom image in Figure 19 we can see a large structure that is alternating between classes 2 and 4. We have previously identified this as a smoke layer which means the entirety of it should be classified as stratospheric aerosol which is class 4.

5 Discussion

Overall, the combination that performed the best was also the simplest. The combination of KMeans clustering and PCA, which can be seen in both Figure 11 and 12, reliably managed to find both ice clouds and smoke layers for a variety of different models.

A reason some of the more advanced methods struggled seems to be an overfixation on noise still present in the images. An example of this can be seen in Figure 13 the classification method differentiates between different parts of the background. This was an issue with many other models and combinations that were not shown in the report. The best explanation we have for this problem is that our preprocessing was subpar. When removing noise we set their values to 0.01 which was probably too high. It would likely have been better to set them to 0 and then set the values removed due to their height to -1.

The argmax classification method performed the worst. This can partially be explained by the low number of channels. As previously mentioned, the number of channels in a semantic segmentation problem is usually in the thousands or even tens of thousands. To achieve a high degree of differentiation the 64 channels that we used are probably inadequate for this method. However, this might not be enough. The paper which achieved success using this method also used it to create pseudo-targets that it incorporated in the training process[5]. This arguably makes the method self-supervised and might be necessary for the efficacy of this method.

Several improvements could be made easily with access to a computer with more memory. For the CNN, increasing things such as the batch size and the number of convolutional layers and channels is all but certain to increase the performance. Additionally, training on more data would also help. Especially, if we started to use color ratio and total

backscattering as inputs. Colour ratio would be especially helpful for differentiating between ice clouds and smoke since it depends on the size of the particles and since ice particles are much larger than smoke particles ($\sim 10\mu m$ as compared to $< 1\mu m$) it should certainly help improve performance.

However, all these combined would require hundreds if not thousands of gigabytes of RAM and it is uncertain how good the results would be. We would venture to say that to truly exceed the capabilities of the SIBYL algorithm the process itself would need to be more advanced. This could be by adopting pseudo-targets and the argmax algorithm like Kanezaki et al.[5] or with a different approach from a newer paper.

Throughout this project, methods have been evaluated primarily by visually comparing the output and the raw data and qualitatively assessing if it is reasonable. When the method is entirely unsupervised there is usually no better option but this introduces bias by itself. What makes the matter worse is that we are using several unsupervised methods at once. The process of semantic segmentation performed in this project can be broken down into four parts: preprocessing, CNN training, dimensionality reduction, and classification. Some of these have metrics from which they can be evaluated but they are all still unsupervised. This means that none of them can be evaluated independently. The only true test is qualitatively judging the final output. Even if the classification method underperforms, it may still be optimal if modifications are made to other components of the process, such as preprocessing or dimensionality reduction techniques.

The evaluation of unsupervised methods is not a new problem. There have been several efforts to circumvent this in various ways and it is not uncommon for studies to completely omitt evaluating their models quantitatively[10].

6 Conclusion

In this study, we have investigated the potential of unsupervised semantic segmentation as a novel approach to address the significant bias present in current methods for classifying aerosols in CALIOP data. Through a comprehensive exploration of various preprocessing techniques, dimensionality reduction methods, and classification algorithms, we have demonstrated promising results in improving the accuracy and reliability of aerosol classification.

Despite the computational constraints imposed by working with limited resources, our research has shown that certain model configurations can effectively identify key features and maintain continuity in segmented images. These findings suggest that unsupervised semantic segmentation holds promise as a viable alternative to traditional classification methods like SIBYL.

The implications of reducing bias in CALIOP data extend far beyond improved classification accuracy. By enhancing our understanding of aerosol radiative forcing and its implications for climate change, this research contributes to the refinement of climate models and the reduction of uncertainties in climate projections.

7 Outlook

In addition to its immediate impact on aerosol classification accuracy, our study paves the way for several exciting avenues of future research and application. Firstly, as computational resources continue to advance, there is immense potential for further refinement and optimization of unsupervised semantic segmentation models, leading to even greater accuracy and efficiency in aerosol classification. Additionally, exploring the integration of machine learning techniques with other sources of Earth observation data could provide a more holistic understanding of aerosol dynamics and their effects on climate. Furthermore, as our understanding of aerosol radiative forcing grows, there is an opportunity to apply these insights to inform targeted climate mitigation strategies and policy decisions. By fostering collaboration between data-driven methodologies and climate modeling, our research sets the stage for continued advancements in atmospheric science and climate research, ultimately guiding more accurate predictions and impactful climate policy interventions.

References

[1] J. Kar, K.-P. Lee, M. A. Vaughan, J. L. Tackett, C. R. Trepte, D. M. Winker, P. L. Lucker, and B. J. Getzewich. Calipso level 3 stratospheric aerosol profile product: version 1.00 algorithm description and initial assessment. *Atmospheric Measurement Techniques*, 12(11):6173–6191, 2019.

[2] Sanja Bauk, Nexhat Kapidani, Zarko Luksic, Filipe Rodrigues, and Luis Júdice Sousa. Review of unmanned aerial systems for the use as maritime surveillance assets. pages 1–5, 02 2020.

[3] Mark Vaughan, Michael Pitts, Charles Trepte, David Winker, Brian Getzewich, Jason Tackett, Xia Cai, Pauline Detweiler, Anne Garnier, Jayanta Kar, et al. Cloud-aerosol lidar infrared pathfinder satellite observations (calipso)-data management system: Data products catalog v4. 95. Technical report, 2023.

[4] Cheng-Jian Lin, Sun, and Wang. Evolutionary-fuzzy-integral-based convolutional neural networks for facial image classification. *Electronics*, 8:997, 09 2019.

[5] Wonjik Kim, Asako Kanezaki, and Masayuki Tanaka. Unsupervised learning of image segmentation based on differentiable feature clustering. *CoRR*, abs/2007.09990, 2020.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Atmospheric Science Data Center. CALIPSO Quality Statements Lidar Level 1B Profile Products. `https://eosweb.larc.nasa.gov/sites/default/files/project/calipso/quality_summaries/CALIOP_L1ProfileProducts_3.00.pdf`, 2006. Version Release 3.00.

[8] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[9] Mark A. Vaughan, David M. Winker, and Chris A. Hostetler. Sibyl : a selective iterated boundary location algorithm for finding cloud and aerosol layers in calipso lidar data. 2003.

[10] Shan Zeng, Ali Omar, Mark Vaughan, Macarena Ortiz, Charles Trepte, Jason Tackett, Jeremy Yagle, Patricia Lucker, Yongxiang Hu, David Winker, Sharon Rodier, and Brian Getzewich. Identifying aerosol subtypes from calipso lidar profiles using deep machine learning. *Atmosphere*, 12:10, 12 2020.

A Appendix

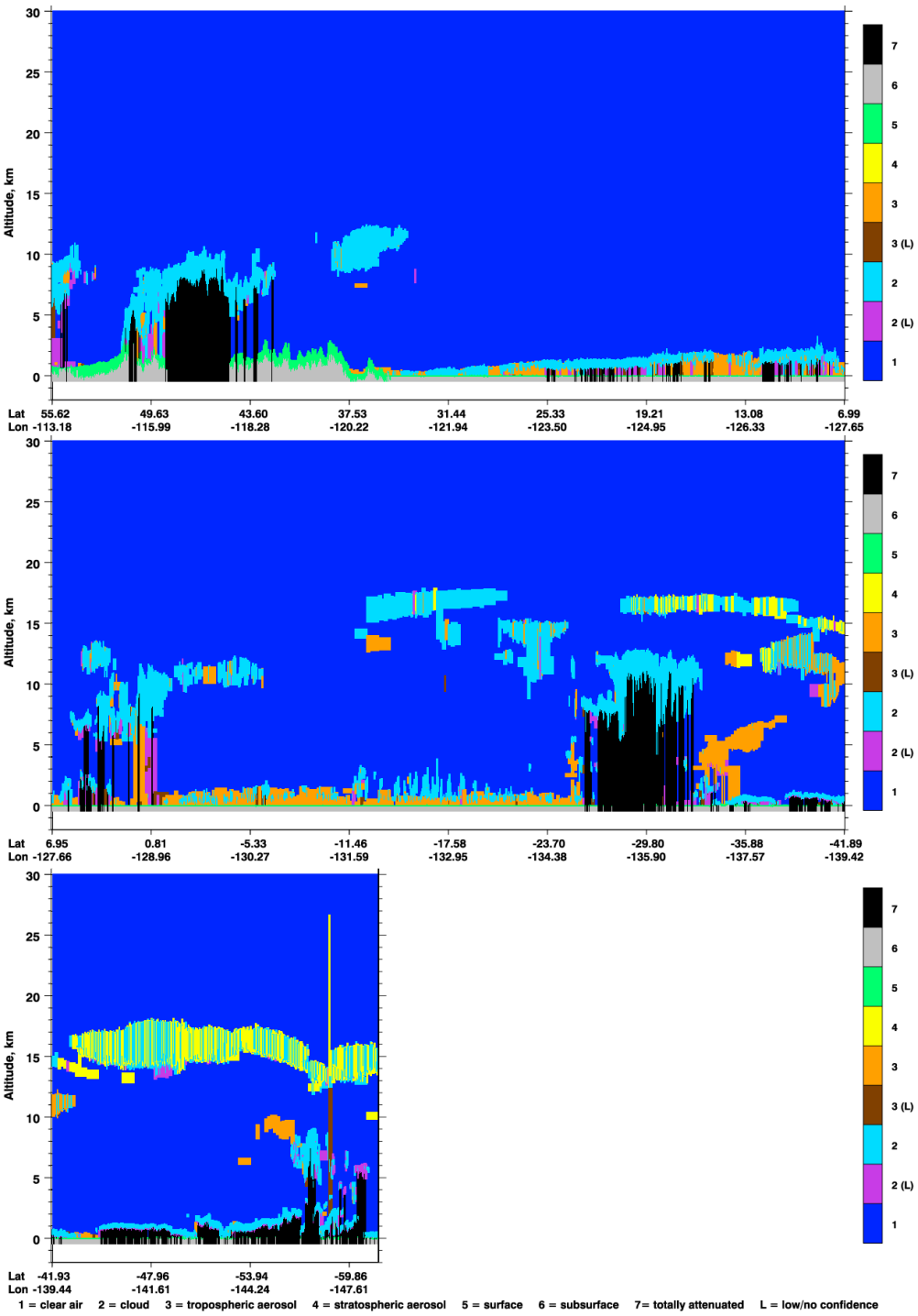

Figure 19: Vertical feature mask created using the SIBYL algorithm. The images correspond to the same orbit as all others, which started at 10:43:03 on the fourth of January 2020. Images taken from: `https://www-calipso.larc.nasa.gov/products/lidar/browse_images/std_v451_showdate.php?browse_date=2020-01-04`

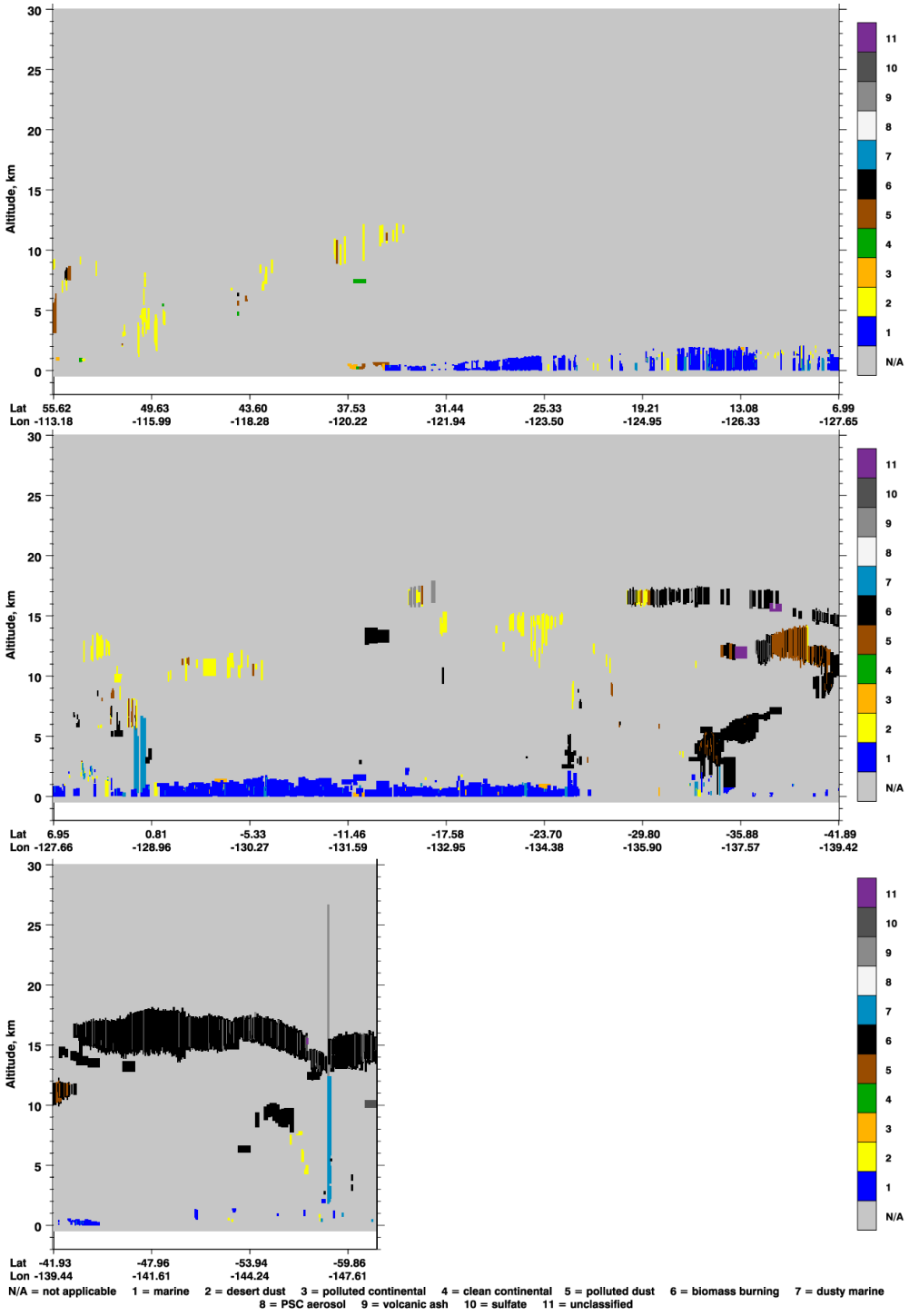

Figure 20: Aerosol subtypes according to the SIBYL algorithm. The images correspond to the same orbit as all others, which started at 10:43:03 on the fourth of January 2020. Images taken from: `https://www-calipso.larc.nasa.gov/products/lidar/browse_images/std_v451_showdate.php?browse_date=2020-01-04`