University of Oxford

Atmospheric, Oceanic & Planetary Physics Department

Using AI to Classify Volcanic Ash in Satellite Observations

Michael Waugh

michael.waugh@lincoln.ox.ac.uk

Summer Project

Supported by the UK Met Office - Oxford Academic Partnership

Supervisors: Andrew Prata & Don Grainger,
andrew.prata@physics.ox.ac.uk/don.grainger@physics.ox.ac.uk

Department of Physics
Atmospheric, Oceanic & Planetary Physics Department
Clarendon Laboratory, University of Oxford, Sherrington Rd, Oxford OX1 3PU

October, 2021

Abstract

This project has investigated the suitability of using machine learning algorithms to determine the presence of volcanic ash clouds using data from the Himawari-8 satellite. Support vector machine (SVM) models were trained on data sets created using an approach combining corrected brightness temperature readings from Himawari-8 data and polygons provided by the Tokyo Volcanic Ash Advisory for ash clouds emitted from Nishinoshima volcano during August and September of 2020. The SVM models with a linear kernel were found to provide accurate estimates of the ash cloud positions in unseen scenes with limited noise which could be mostly removed in post-processing. The models were found to perform more robustly and accurately when trained on a larger data set containing data points from a wider variety of scenes. The linear kernel model identified the 8.60 µm, 9.63 µm and (particularly) 12.35 µm Himawari-8 channels to be significant in determining the lack of ash, while the 11.20 µm and 10.45 µm channels were significant in determining the presence of ash. Decision tree and k-nearest neighbour models were briefly investigated and it was found that decision tree classifiers produced results of similar quality to SVM models on the data sets used; however, k-nearest neighbour models took too long to run to be viable. The SVM models were also run over a separate ash cloud from the Raikoke (Russia) eruption observed on June 22, 2019, where it was able to predict the location and shape of the ash cloud with promising accuracy when inspected visually.

Contents

Chapter 1

Introduction

1.1 Volcanoes and Volcanic Ash

Volcanic ash contributes to a substantial impact on the aviation industry and may influence climate when significant amounts of SO_2 are realised during an eruption.

For the aviation industry, ash can melt onto a jet engine turbine and cause failure, because the melting point of ash is around 1100 K while jet engines operate at around 1400 K. In addition to engine damage, volcanic ash can damage cockpit windows and other airplane surfaces, obstruct the pitot-static system, and be ingested into the air conditioning and cooling systems [1].

Climate-wise, volcanic ash particles are assumed to have a short atmospheric lifetime, although they may last for about three months and therefore their radiative influence could affect surface cooling [2]. After an eruption, the initial SO_2 lifetime is determined by the SO_2 uptake on ash [3]. Sulfur dioxide reacts with H_2O in the atmosphere to produce sulphate aerosols such as H_2SO_4 which last for 1–3 years when injected into the stratosphere and can result in a global surface cooling due to the reflection of shortwave radiation [4, 5]. Furthermore, ash provides a source of cloud condensation nuclei and ice nuclei, increasing the number of cloud droplets which modifies cloud lifetime and precipitation processes [6]. It can also serve as a source of soluble iron to the ocean, increasing marine primary production when iron is the limiting nutrient, reducing concentrations of atmospheric CO_2. Visibly obvious effects of volcanic ash from eruptions is that its deposition on snow and ice

surfaces can result in surface darkening, snowmelt, and snow insulation (for a thick ash deposit) [7].

1.2 Machine Learning Basics

A machine learning algorithm is an algorithm that is able to learn from data. An algorithm is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [8]. Machine learning tasks are usually described in terms of how the system should process an example, which is a collection of features that have been quantitatively measured from some object or event that we want the system to process.

To create a machine learning model, a chosen algorithm must be trained on a data set, defined as the training data set. The training involves fitting a model in such a way as to minimise a cost function which specifies the performance of the algorithm. The algorithm is then tested on a test data set which has not been included in the training data set. The goal of a machine learning algorithms is to minimise its generalisation error, meaning the algorithm performs as accurately as possible on data not seen in the training data. This should align with the error seen when forming predictions on the test data set, however, does not correspond to the error seen when performing predictions on the training data set. Minimising training data at the expense of the generalisation or test accuracy is called over-fitting and occurs when the model is matched too closely to the training data.

Learning algorithms can be roughly split into two types - supervised learning and unsupervised learning. Supervised learning algorithms are applied to a data set containing examples with a set of features, but each example is also associated with a label or target. For example, a data point in a data set could be an image, with each pixel representing a feature and the image being labeled as containing a dog. Example supervised learning algorithms are Support Vector Machines, K-Nearest Neighbour and Naive Bayes. Unsupervised learning algorithms experience just a data set containing features, with no labels given. These algorithms attempt to learn patterns from the data set themselves. Example unsupervised learning algorithms are Clustering, Principal Component Analysis and Autoencoders.

We typically represent a data point as a vector $\boldsymbol{x} \in \mathbb{R}^n$, where each entry x_i of the vector is another feature. One common way of describing a data set is with a design matrix $M \in \mathbb{R}^{m \times n}$ with each row representing one of m data points and each column corresponding to one of n features for our data set. For a supervised learning algorithm we also provide a vector of labels $\boldsymbol{y}$, with y_i providing the label for example i [8].

1.2.1 Support Vector Machines

One of the most influential approaches to supervised learning is the Support Vector Machine (SVM). This model is driven by a linear function $\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}$ for input vector $\boldsymbol{x}$ where the feature weights vector $\boldsymbol{w}$ and bias b are features to be determined in the training process. This model is used for classification, where the positive class is present when $\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}$ is positive or equal to zero, and the negative class is present when $\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}$ is negative [8].

We can describe this process as constructing a hyper-plane or set of hyper-planes in our data-space to divide our classes in a classification task. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class, called the support vectors, since in general the larger the distance the lower the generalisation error of the classifier when being tested on a test data set [9].

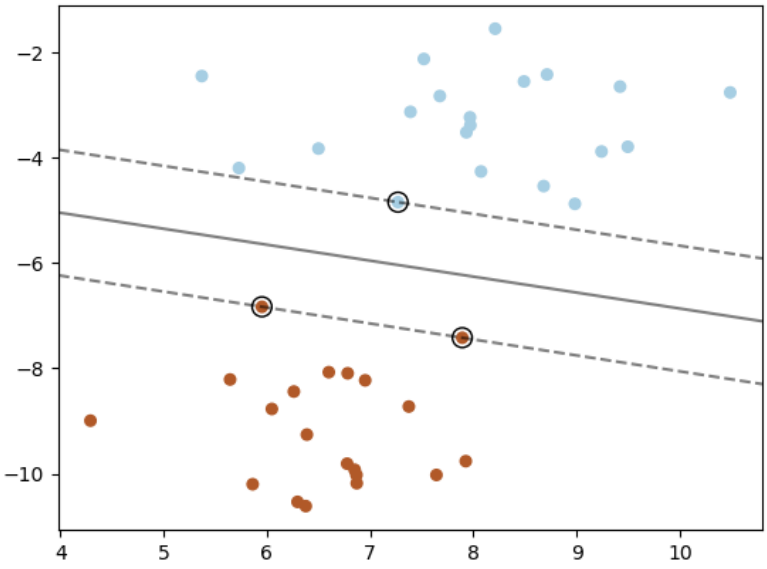


Figure 1.1: The hyperplane given by a support vector machine separates the two classes in a classification task [9]

One key innovation associated with SVMs is the kernel trick. Kernels allow us to add more features to a data set so that the data becomes linearly separable. This means that a data set is transformed to a new space and learning a linear model in this new transformed space. A mathematical treatment can be read in most good machine learning books which refer to SVMs, such as [8]. The kernel trick then enables us to learn models that are nonlinear using convex optimization techniques which are guaranteed to converge efficiently.

Chapter 2

Methodology and Investigation

2.1 Case Studies

2.1.1 The Nishinoshima Eruption

Japan's Nishinoshima volcano is located about 1000 km south of Tokyo in the Ogasawara Arc [10]. For the purposes of this project work focused upon eruptive activity spanning March-August 2020. During this period, major lava flows covered all sides of the island, with higher levels of activity during late June and early July. Ash emissions increased significantly during June and produced dense black ash plumes that rose up to 6 km altitude in early July. Explosive activity produced lightning and incandescent jets that rose 200 m above sea level and large bombs that fell to the base of the pyroclastic cone. Lava flow activity diminished at the end of July. Ash emissions decreased throughout August. Sulfur dioxide emissions were very high during late June through early August, producing emissions that drifted across much of the western Pacific region. [10].

Figure 2.1: Image taken by the Japan Coast Guard when conducting an overflight of Nishinoshima on 9 March 2020 [10]

2.1.2 The Raikoke Eruption

Raikoke is a Russian volcano situated north of Japan. On June 21, 2019, at around 18:00 UTC, the volcano erupted for the first time since 1924, with volcanic ash and gases being released and moving east as they were pulled into the circulation of a storm in the North Pacific. Significant volcanic activity lasted until June 23, 2019 with smaller amounts of ash, gas and steam continuing to be released until 1 July, 2019 [11], [12]. This eruption featured significant cloud cover, differing from the atmospheric environment associated with the Nishinoshima eruption case study.

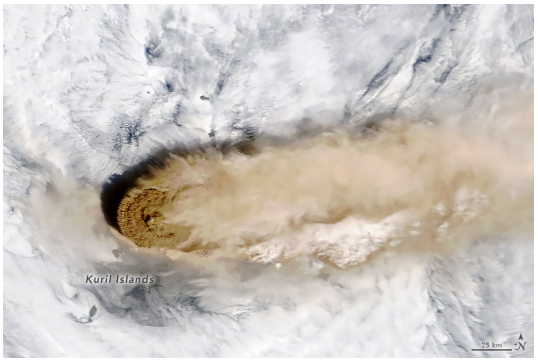


Figure 2.2: Raikoke erupting on June 22, 2019 [11].

2.2 Volcanic Ash Data Set Creation

To train and test a machine learning algorithm a benchmark description of ash location is required. In this project, two different ash location descriptors were conflated to produce final labelled data sets. The first finer descriptor is to use data from the Himawari-8 satellite to calculate a corrected brightness temperature difference (BTD) for pixels in each satellite image, while the second coarser descriptors are volcanic ash polygons provided by the Tokyo Volcanic Ash Advisory Center (VAAC) [13].

2.2.1 Himawari-8 and the Corrected Brightness Temperature Differences

Himawari-8 is a Japanese weather satellite that was launched in October 2014 and is operated by the Japan Meteorological Agency. To identify ash the Advanced Himawari Imager (AHI) instrument can be used, which has 16 channels operating at different wavelengths with resolutions ranging from 0.5 km to 2.0 km (at nadir). More information on these channels is given in Table 2.1 [14]. Specifically, a corrected BTD is to be calculated which uses bands 14 (centered on 11.20 μm) and 15 (centred on 12.35 μm). With these two channels we calculate the BTD by subtracting the brightness temperature for band 15, T_{12}, from band 14, T_{11}. This can then be

corrected for water vapour by applying the Yu et al. semi-empirical water vapour correction:

$$\Delta T_{11-12} = \exp\left(\frac{6}{320}T_{11} - b\right). \tag{2.1}$$

A corrected BTD (ΔT_{11-12}) of less than zero is indicative of the presence of ash [15]. The free parameter b essentially determines the degree to which the BTD is corrected for water vapour was set to be 4.5 for this project. The free parameter can actually be set to an optimal value for each scene as described in [15]; however, in this project it was kept constant. Both bands 14 and 15 have resolutions of 2.0 km at nadir so this produces an image of resolution 2.0 km pixels which are designated to either contain ash or to not contain ash. Visual representation of all the corrected BTDs can be seen in Sect. A.1, with an example shwon in Fig. 2.3. All AHI channels are used as an input for the machine learning algorithms used in this project; however, since different channels have different resolutions those of higher resolutions are re-sampled to match the lower resolution channels by averaging pixel values.

Band	Central Wavelength	Bandwidth	Resolution
1	455 nm	50 nm	1.0 km
2	510 nm	20 nm	1.0 km
3	645 nm	30 nm	0.5 km
4	860 nm	20 nm	1.0 km
5	1610 nm	20 nm	2.0 km
6	2260 nm	20 nm	2.0 km
7	3.85 μm	0.22 μm	2.0 km
8	6.25 μm	0.37 μm	2.0 km
9	6.95 μm	0.12 μm	2.0 km
10	7.35 μm	0.17 μm	2.0 km
11	8.60 μm	0.32 μm	2.0 km
12	9.63 μm	0.18 μm	2.0 km
13	10.45 μm	0.30 μm	2.0 km
14	11.20 μm	0.20 μm	2.0 km
15	12.35 μm	0.30 μm	2.0 km
16	13.30 μm	0.20 μm	2.0 km

Table 2.1: Characteristics of the AHI Instrument.

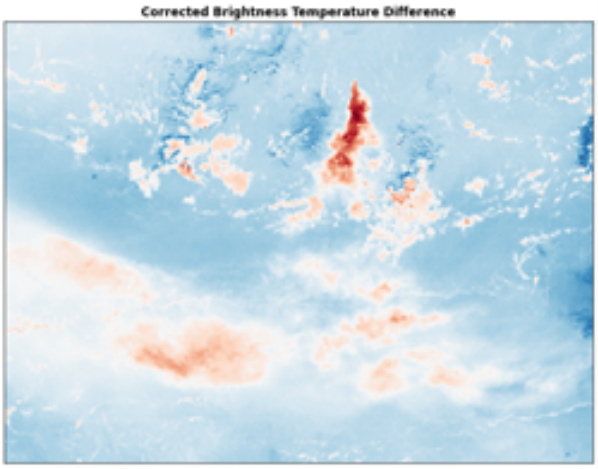


Figure 2.3: Corrected BTDs for data set from 06:00 UTC on 01/08/2020. Red pixels indicate 'ash' while blue pixels indicate 'not ash'

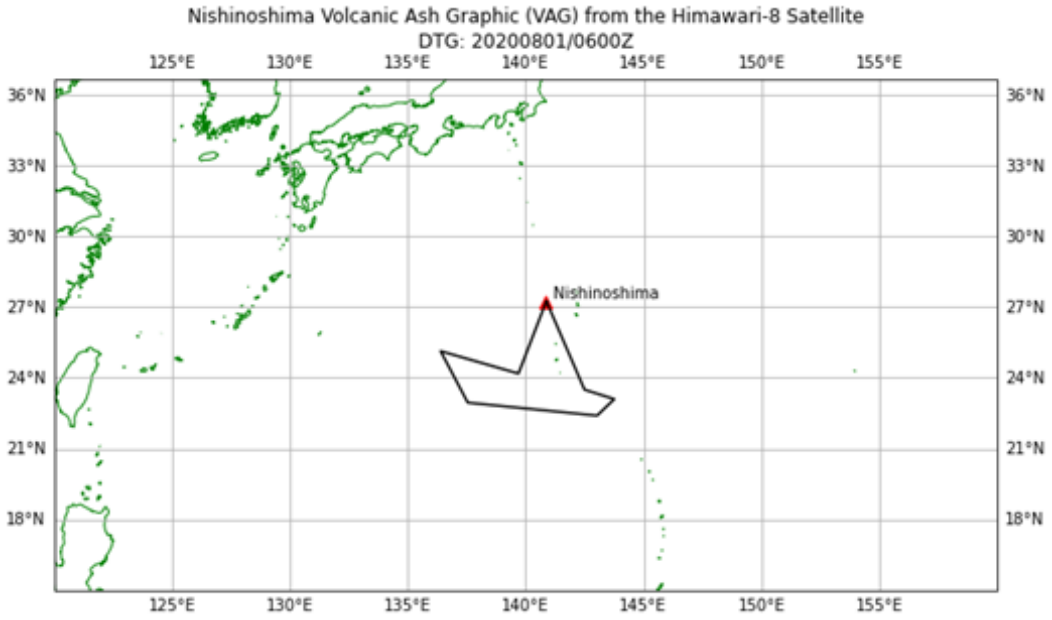
2.2.2 Tokyo VAAC Polygons

The Tokyo VAAC produce volcanic ash graphics which contain a polygon indicating the current location of ash, as well as ash location forecast polygons at 6, 12 and 18 hours after the known ash position. The known polygons are based on satellite observations, model guidance, ground and air-based observations (when available) and human expert interpretation. To produce labelled data sets for this project, the polygon corresponding to the observed location of the ash cloud is used in combination with the water vapour-corrected BTDs. This observed location polygon has its vertex coordinates provided by the VAAC so can be easily accessed. These polygons represent a coarser view of the ash location than the corrected BTDs from the Himawari-8 data; however, they are useful because they can be combined with the finer automated approach to remove false positives which commonly appear when a simple BTD threshold is used to detect ash. Examples can be seen in the volcanic ash graphics included as part of Appendix (Sect. A.1).

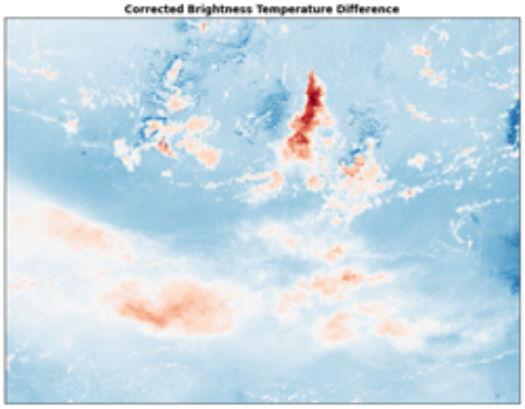
2.2.3 Combining Approaches to Create a Labelled Data Set

To produce a data set combining the automated corrected BTD approach with the human VAAC polygon approach, each pixel has its corrected BTD calculated. If this value is less than zero and the pixel is located within the VAAC polygon it is designated as containing ash and all other pixels are designated as not containing ash. This data set consists of a regional subset of the full disk satellite image, but depending on the chosen image size each data set created this way could contain hundreds of thousands of pixels. Multiple observations in time (every 10 minutes) can also be considered since Himawari-8 is a geostationary satellite, which can be

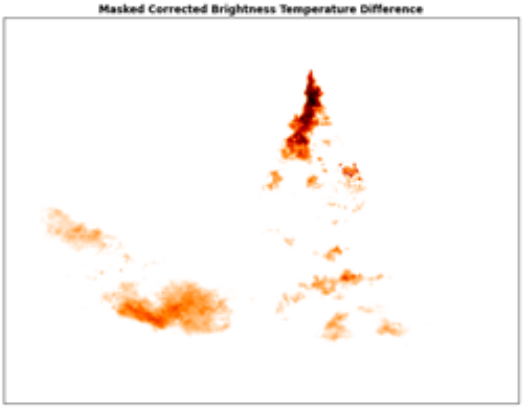
combined to create larger data sets which span a greater variety of conditions. Examples of these final data sets can be seen in Sect. A.1, with an example shown in Fig. 2.4.



(a) Volcanic Ash Graphic (VAG) showing the Tokyo VAAC polygon.



(b) Visualisation of the corrected brightness temperature (BTD) for each pixel, with red demonstrating the presence of ash and blue demonstrating a lack of ash.



(c) Final data set created by combining the VAAC polygon and corrected BTD approaches.

Figure 2.4: Creating the data set for 06:00 UTC on 01/08/2020

2.2.4 Data Sets Used

For this project four scenes have been used to create training data sets for the Nishinoshima eruption. Two of these are nighttime scenes and two are daytime scenes and are shown in Figs. A.1–A.4. It should be noted that the scene shown

in Fig. A.4 has produced a nonphysical-looking ash cloud. This is due to an over-correction from using the water vapour correction with the b parameter kept fixed at 4.5. Instead, b should be varied according the scene to account for the variation in water vapour in time and space but since it has been kept fixed for the other data sets it may be interesting to evaluate the performance of the algorithm trained on the other three on this fourth more difficult scene.

2.3 Approach for Training and Testing the Algorithm

A labelled data set can be used to train and test a machine learning algorithm. This project used the scikit-learn library to implement machine learning algorithms and generally a simple test/train split has been used to divide a data set into a training data set and a test data set because it is fast and convenient. In some cases, where it shall be stated, a repeated stratified k-fold cross-validation approach was implemented, where the data set is divided into k equal sub-data sets with equal proportions of 'ash' and 'not ash' pixels before cycling through training on all but one fold and testing on the remaining one then repeating the process a given number of times with different splits to produce a final best algorithm. Since this process has multiple testing phases it provides a model accuracy mean and standard deviation but does take longer and is less flexible to user-implemented variation since it uses an entire data set for training and testing at once.

In general, learning algorithms benefit from normalisation of the data set so this was included whenever a learning algorithm was being used [9]. In particular, the StandardScaler utility class provided by scikit-learn's preprocessing module was used, which re-samples the data set to have zero mean and unit variance [9].

2.4 The Support Vector Machine

The built-in scikit-learn Support Vector Classifier (SVC), which operates as a SVM being used for the task of classification, has been the main focus of this project. It is a simple, versatile and widely used machine learning algorithm which is effective over a range of data set sizes, making it ideal for an investigative project. In this project different SVCs have been trained on a variety of situations to build an understanding of how different training data sets affect the performance of the algorithm and it's generalization ability.

In training, the different SVCs considered pixels individually, where they were given data for all sixteen channels from Himawari-8 for that pixel alongside a label of whether the pixel contained ash or not. It should be noted that the bands with resolutions of 0.5 km and 1.0 km were regridded to a resolution of 2.0 km by averaging over the smaller pixels. The algorithm could therefore learn which channels

were important features from the satellite-measured brightness temperatures and reflectances. To test the trained algorithm, pixels were considered and classified individually. The data sets used for testing and training could range from including pixels from as little as a subset of one image or up to the entirety of multiple images to allow for variation in testing the algorithm. Since images could be taken at different times of day and in different atmospheric conditions this could result in data sets containing a variety of different ash conditions.

It should be noted that an important tool in evaluating the performance of our algorithm was to visually inspect the predictions. False positives were often found to appear around the edges of the ash cloud boundary when comparing the classifier output to training data set (i.e. corrected BTD and VAAC approach). This could indicate that the algorithm is producing a more sensitive detection scheme than the combined BTD-VAAC approach, which would be a positive outcome. Furthermore, the size of the images tested over was often large compared to the quantity of volcanic ash pixels present in the image so even algorithms with a very low rate of false positives could produce a large number of false positives proportional to the quantity of ash pixels. Numerically this looks challenging but by visually inspecting our predictions we can determine the nature of the false positives. If they represent significant distortions of the ash cloud, or additional regions of ash, this would be an issue however they may simply be multiple small regions of background noise which could be removed in post-processing.

When interpreting the model predictions visually, care must be taken when looking at scenes involved in the training process. Pixels from these images have been involved in training the algorithm and cannot reliably be used in testing, so images involved in the training may be predicted more accurately than images not involved in the training. Ideally only images not included in the training data should be visualised, however when the subset of data included in the training data from an image is small, the model predictions can nonetheless provide insight into the algorithms behaviour.

Bearing this in mind, to post-process our predictions the 'open' morphological transformation with a 3×3 kernel built into the OpenCV package [16] was used when stated. This effectively removed objects determined to be ash which were less than 3×3 pixels in size. This was mainly done to more easily inspect the performance of the algorithm at determining the location of the large ash clouds coming directly from the volcano and the nature of these small regions of noise was not investigated.

2.4.1 Linear vs. Non-linear Kernel and Hyperparameter Setting

To optimise the performance of a machine learning algorithm the hyperparameters associated with the algorithm should be set to some optimal values. The optimal values depend on the data being considered and can be found methodically using

a variety of models, such as a grid search over a range of hyperparameter values. Grid search is often relatively slow and in this project Bayes optimization using a cross-validation approach was used to gain an idea of suitable hyperparameter values, making use of the BayesSearchCV class built into scikit-optimize [17]. The optimal hyperparameter values depend on the data set and given that optimization methods are relatively slow it was impractical to optimize the values for every trial run. Instead, a set of 'good' hyperparameters guided by using the BayesSearchCV class over a variety of data sets were used.

The hyperparameter from the scikit-learn package for the Linear SVC which we shall vary is the parameter C which determines the degree of the soft boundary, so a large value of C means less tolerance of outliers while low C means more tolerance. For the SVC (not necessarily with a linear kernel) there is the additional hyperparameter, γ, which is similar to C and is a regularisation parameter which controls the degree of under-fitting and over-fitting. The 'degree' hyperparameter controls the degree of kernel used, of which a higher value results in a more flexible decision boundary. For both the SVC and Linear SVC the optimal value of C was found to be 10000 by the BayesSearchCV class, then for the SVC the optimal values of γ and the kernel degree were found to be 0.16 and and 4, respectively.

In this investigation we wanted to see whether a linear or non-linear kernel performs better. A non-linear kernel adds features by applying a non-linear function to the data points to be better divided by the model hyper-plane; however, the performance of the kernel settings should be measured against one another to determine which is truly best.

In order to compare linear and non-linear SVCs, a data set was created using all of the 'ash' pixels from three scenes shown in Fig. A.1, Fig. A.2 and Fig. A.3 and ten times that amount of 'not ash' pixels from the same scenes. This data set was then split into 80% training data and 20% test data before being used to train a linear kernel SVC and a non-linear kernel SVC. These models were then run over the entirety of the scenes included in the training data and then the resulting classifiers were applied to an entirely new scene. The results for these are provided in Sect. B.1. The non-linear kernel SVC can be seen to fit better to the known 'ash' positions for the scenes which were included in the training data than the linear kernel SVC (compare Fig. B.5 to Fig. B.1); however, for the fourth scene (which was not included in the training data) the non-linear kernel SVC has not predicted a single correct ash pixel, as seen in Fig. B.8. On the other hand, the linear kernel SVC has produced predictions shown in Fig. B.4 which appear to somewhat align with the Tokyo VAAC polygon and true colour image (Fig. 2.5) for that scene, despite the fact the water vapour correction appears to have over-corrected for water vapour in this scene.

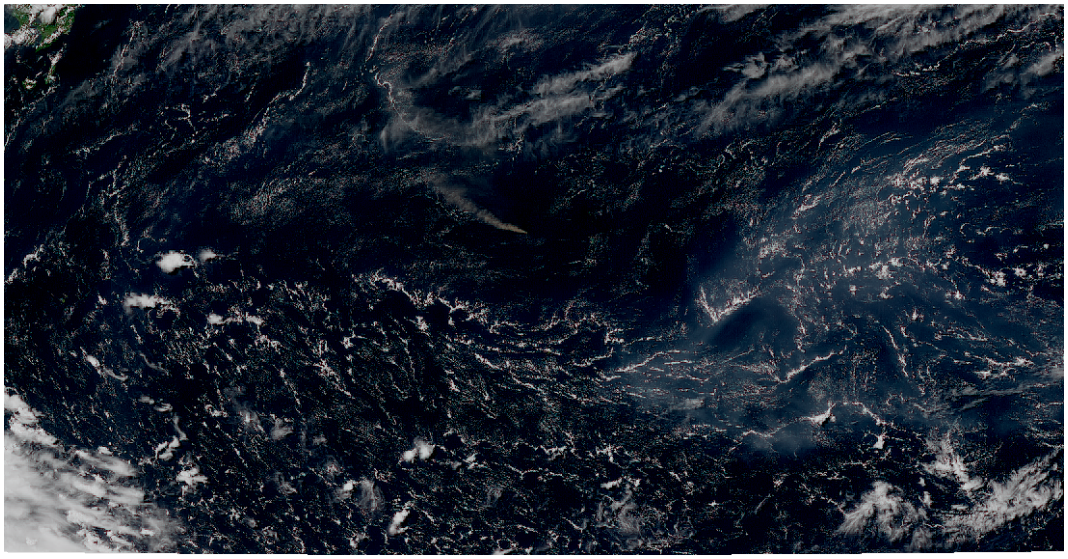


Figure 2.5: True colour image representing the data set A.4.

These results demonstrate that the non-linear kernel can indeed fit training data very well; however, it may produce a model which fails to generalise well on unseen data. This effect may have been mitigated by changing the hyperparameters of the models; however, this was not explored fully in this investigation. Alternatively, the linear kernel SVC appears to perform somewhat less precisely, but generalises far better and produces useful results even in conditions with differing concentrations of water vapour when little training data is used. Given this property, unless otherwise stated a linear kernel was chosen for SVC algorithm used in this project.

2.4.2 Quantity of Training Data

A fundamental part of any machine learning algorithm is the training process. An increase in the quantity of training data should generally increase the accuracy of a machine learning process to a point; however, this has caveats which should be investigated.

To investigate the impact of the quantity of training data on the algorithm's performance, a subset of pixels were used to train a SVC on a single scene (Fig. A.1). Varying the size of the training subset and its ratio of 'ash' to 'not ash' pixels allows changes in the training error to be investigated. A SVC was trained using a data set containing 50% 'ash' pixels in the image and 50% 'not ash' using a test/train split of 20% training data and 80% test data. We then compared this with a SVC trained on the same proportion of 'ash' to 'not ash' pixels but with 80% training data and 20% test data. The impact of increasing the proportion of training data can be seen by comparing Figs. B.11 with B.9. There is relatively little difference between predictions, although particularly in the top left quadrants of Fig. B.11

and Fig. B.9 we can see fewer false positives in the model with a larger training data set. Testing the two SVCs on a scene completely separate from the training data produces Fig. B.12 (with scikit-learn model score over the scene of 0.9757) and Fig. B.10 (with scikit-learn model score over the scene of 0.9812), which show that the increased quantity has increased the model score on the scene included in the training data. This effect can be somewhat seen in that there are fewer ash points missed in Fig. B.10.

Considering the same scene, Fig. A.1, a data set can be created with a 1:10 ratio of 'ash' to 'not ash' pixels. This produces a much larger data set than considered in the previous paragraph, although it is very unbalanced. The effect of this is useful to investigate because a relatively small proportion of a given satellite image contains ash, so if increasing the quantity of 'not ash' pixels improves the model performance then the data sets can easily be expanded. Splitting the unbalanced data set into 80% training data and 20% test data produces predictions seen in Figs. B.13 and B.14. It can easily be seen that this reduces the number of additional ash pixels predicted in the image included in the training data when compared to the predictions in Figs. B.9 and B.10 but increases the number of ash pixels missed by the algorithm. This effect can be seen even more clearly when considering the scene not included in the training data (Fig. B.14). Here there are relatively few false positive results but a very large number of false negatives. Since the algorithm has been trained on a much larger number of 'not ash' points, the algorithm benefits more from producing a hyper-plane closer to the boundary of the 'ash' data points when minimising the cost function in the training process since there is a much larger number and variety of 'not ash' points. Nonetheless it can be seen that the algorithm appears to have become better at classifying 'not ash' than the number of 'ash' predictions it incorrectly misses. This seems logical since 'not ash' covers a wider range of scenarios than 'ash' and therefore it should take a larger amount of 'not ash' data to accurately describe all atmospheric conditions.

2.4.3 Variety of Training Data

The Himawari-8 images can be a taken under a wide variety of atmospheric conditions. Examples of these conditions include variation in surface temperature, the presence of clouds, amount of water vapour, aerosol and the type of surface. Another consideration is how the information content changes for day vs. night measurements (i.e. with and without daytime reflectance information). The variety of atmospheric conditions seen in Himawari data suggests that a wide variety of training data is required to produce an accurate algorithm.

To test the impact of altering the variety of training data on the SVC algorithm's performance, a data set was created using all of the 'ash' labelled pixels and an equal number of 'not ash' pixels from a single scene (shown in Fig. A.1). A SVC was then

trained on 75% of the data set and tested on the remaining 25%, before also being tested on two other scenes not included in the training or testing data sets. The results of this are visualised in Sect. B.3.1. Now consider a data set created using all of the 'ash' pixels and an equal number of 'not ash' pixels from all three of these images, before being split into 25% training data and 75% testing data. This can be used to train another SVC, which was trained on a similar amount of training data as the one trained on just one image but with data from three scenes, providing a training data set with a wider range of variety. The results of testing this second SVC on the images are shown in Sect. B.3.2. Comparing the predictions from both SVCs it can be seen that the one trained on a wider variety of scenes actually produces a model score that is very close to or slightly below that of the SVC trained on just a single image. However, inspecting the predictions we see that the number of false negatives is greatly decreased in the images not included in the training data for the SVC trained on just a single image, while the number of false positives has increased in the form of edge effects around the cloud or as small regions of background noise. The scenario of fewer false negatives at the cost of increased edge effects and noise is generally preferable because it is possible that the edge effects do indeed represent true classifications of ash missed by the combined corrected BTD and VAAC approach, while noise can be removed in post-processing.

2.5 Other Algorithms

In this project we also considered the ability of two other machine learning algorithms (i.e. decision trees and k-nearest neighbours) to classify ash and are briefly described here. Decision tree classifiers consist of 'nodes' which lead to 'branches' and eventually 'leaf nodes'. Nodes are points at which a test is done and the result governs which branch should be taken to the next node. This continues until a leaf node is reached, which predicts the class the decision tree input belongs in. K-nearest neighbour classifiers simply consider an input and find which training data point the input matches most, then classifies the input as the same class as the training point.

Decision trees can be interpreted relatively easily by viewing the tree as a whole and can provide input into how the model is working, allowing for the physics behind the decision to be interpreted without too much difficulty. However, they can overfit data sets if they are not managed correctly. Limits can be set on the depth of decision trees and this helps to reduce overfitting.

K-nearest neighbour classifiers are simple to understand, quick to train and effective in a variety of machine learning tasks; however, the time taken to classify inputs is generally large depending on the method used by the algorithm.

Chapter 3

Results

3.1 Linear Kernel Support Vector Machine

As described in Sect. 2.4, an effective SVC for classifying volcanic ash clouds requires a large quantity and wide variety of training pixels. Given this we can consider full segments of the Himawari-8 scenes around Nishinoshima.

To test if the linear SVC algorithm could classify an entire segment of Himawari data, a data set comprising all ash pixels from three scenes and twenty times that number of 'not ash' pixels from the same scenes was created before being split into 80% training data and 20% test data. This data set is made up of a variety of scenes and as large a quantity of data as is realistically usable from each image, with a heavy weighting towards 'not ash' pixels with the aim of reducing false positives over the large segments while keeping the shape and size of the predicted ash cloud accurate. A linear kernel SVC can be trained and then tested over the full three scenes used in the training data and an unseen fourth scene, before the predictions are post-processed by removing all regions predicted to be 'ash' smaller than 3×3 pixels using the morphological filter from OpenCV. This produces the results seen in Sect. B.4. Explicitly, the SVC is trained on $384,679$ data points, of which $17,998$ are ash pixels. Each segment is then $2,530,000$ pixels of which only $5,000$-$10,000$ are ash. The model predictions can be seen to closely agree on this scale with the true values found by the combined corrected BTD and VAAC approach. Some false positives are seen although they are often removed effectively

by post-processing, leaving a precise and accurate prediction of the ash cloud location. The noise appears to be found predominantly in the region of the major ash cloud, perhaps indicating smaller ash clouds becoming more diffuse further from Nishinoshima; however, the characteristics of this noise are not considered in this project. In the fourth image not seen by the SVC in the training process, the model predicts an ash cloud which appears to closely match the VAAC polygon and true colour image, despite the fixed variable corrected BTD method used for the training data producing an nonphysically-shaped ash cloud (see top panel of Fig. B.24).

An advantage of using a SVC with a linear kernel is that the coefficients of the model can be meaningfully read, which tells us which features are important. For the SVC trained and tested above the coefficients are shown in Fig. B.25. These coefficients are the values from the weight vector $\boldsymbol{w}$ in Sect. 1.2.1. Therefore, a more positive coefficient corresponding to a particular feature means that coefficient contributes more to the model classifying the pixel as 'not ash' than 'ash' for a positive feature reading, whereas a more negative coefficient corresponding to a particular feature means that coefficient contributes more to the model classifying the pixel as 'ash' than 'not ash'. Now (for data already scaled by the StandardScaler) the (uncorrected) brightness temperature would be recovered by simply having the coefficients for B14 (11.2 µm) and B15 (12.35 µm) as -1 and 1 respectively, while all the others are zero. Instead the model weights the coefficients for these features differently and additionally considers B11 (8.6 µm) and B12 (9.63 µm) as important for classifying 'not ash' and B13 (10.45 µm) as nearly as important as B14 (11.2 µm) for classifying 'ash'. Volcanic ash particles exhibit a strong absorption spectra near wavelengths of 9.5 µm due to the high silica component of ash particles. Due to this absorption peak, thermal infrared radiation coming from the surface will be more strongly absorbed at wavelengths near 10 μm and 11 μm than it will at 12 μm, meaning that brightness temperatures will be lower for B13 and B14 than for B15, so it is not surprising bands 13, 14 and 15 are important for ash detection. It is interesting that the 9.63 µm band appears to be important for classifying pixels as 'not ash' when considering that ash absorbs strongly near this wavelength; however, it's possible that ozone absorption is obscuring ash detection in this wavelength region.

If the algorithm is retrained with the same size and proportion of training/testing data as before, then another coefficient plot can be made, shown in B.26. As the training and testing pixels are selected randomly, there is a degree of variability amongst various channel weights despite their proportions being kept constant. For example, B13 has become even more important than B14 for classifying 'ash' when the algorithm was retrained. This degree of variability shows that even over just these three scenes there is enough random variability for the algorithm to vary somewhat in training. This suggests the algorithm would benefit from an even

greater amount of training data, and if additional scenes were used then the increased variety of training data would likely benefit from an even greater number of pixels being considered too.

3.1.1 Testing on the Raikoke Eruption

To investigate the generalisation ability of the SVM models developed, tests can be done on other volcanic eruptions. The Raikoke eruption considered is from around 13 months before the Nishinoshima eruption and located at a different position in the Pacific ocean, so the situation should be somewhat different and provide guidance on whether or not the model shall generalise well to other eruptions when trained on just data from the Nishinoshima eruption.

The following results are shown for a SVM model trained on 80% of the 'ash' pixels from the first three data sets in A.1 and 20 times that number of 'not ash' points from the same data sets. This model was applied to a daytime scene of the Raikoke eruption shown in Fig 3.1 to produce the results visualised in Fig 3.2. Visually comparing the true colour image with the predictions we see that the prediction appears to detect the visible ash trail. This initial test suggests the model is capable of generalising to unseen eruptions, in this case without even expanding the training data set; however, a more in depth analysis would be required to investigate the full accuracy of the predictions.

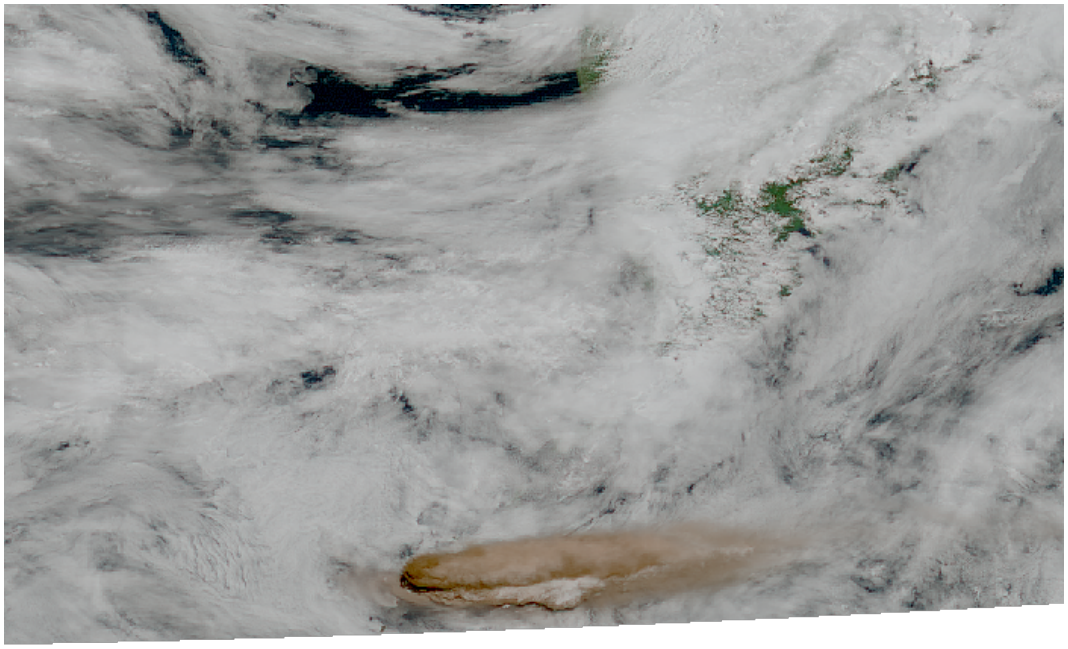


Figure 3.1: True colour image produced using data from Himawari-8 of the Raikoke eruption at 02:00UTC on 22/06/2019.

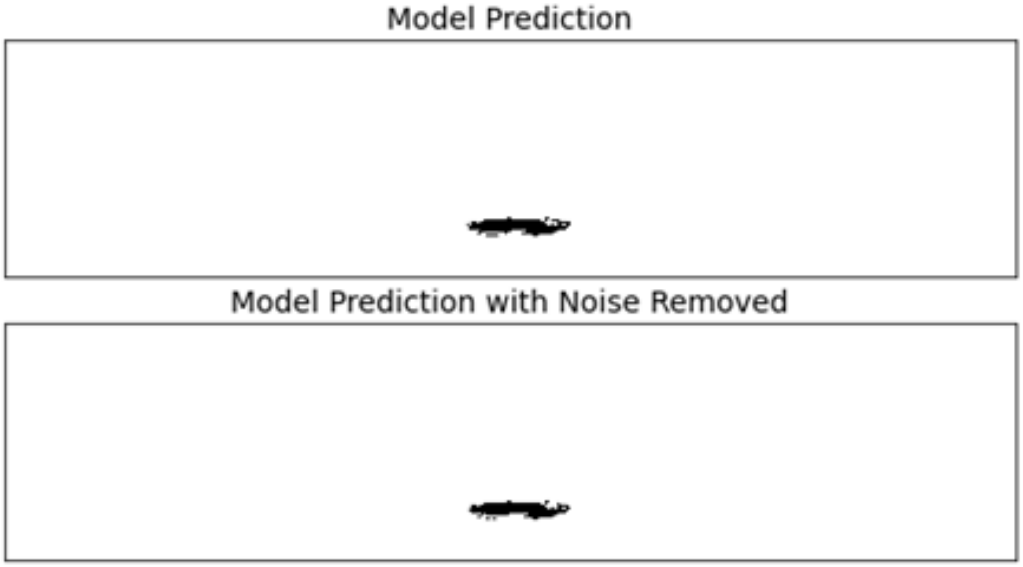


Figure 3.2: Linear kernel SVM prediction using data from Himawari-8 of the Raikoke eruption at 02:00UTC on 22/06/2019.

3.2 Testing Other Algorithms

We now briefly test other models to see how algorithms besides the SVM perform. These models are being run with their default settings from the scikit-learn package so their results may be improved by tuning the models; however, since this is more of a proof-of-concept test we have left this as work to be undertaken in the future. Both classifiers shall be trained on the same data set of 80% of the 'ash' pixels from the first three scenes shown in Sect. A.1 and 20 times that number of 'not ash' pixels from the same three scenes.

The decision tree classifier is quick to train and test on, similar to the SVM models. Testing the classifier on the first scene in Sect. A.1 produces the results shown in Fig. 3.3, which are comparable to the equivalent predictions by the SVM shown in Fig. B.21. This suggests that the decision tree model would be a viable approach for classifying ash pixels; however, it would require further investigation to find how the classifier generalises.

The k-nearest neighbour classifier is very quick to train, since training consists of just committing the training data to memory. However, testing this classifier takes so long that a single frame of ash predictions could not be made. Since making predictions with the model scales with the quantity of training data, dimensionality of the input data and the number of input points, this model does not appear to be a viable approach for creating an 'ash' classifier for geostationary satellite data.

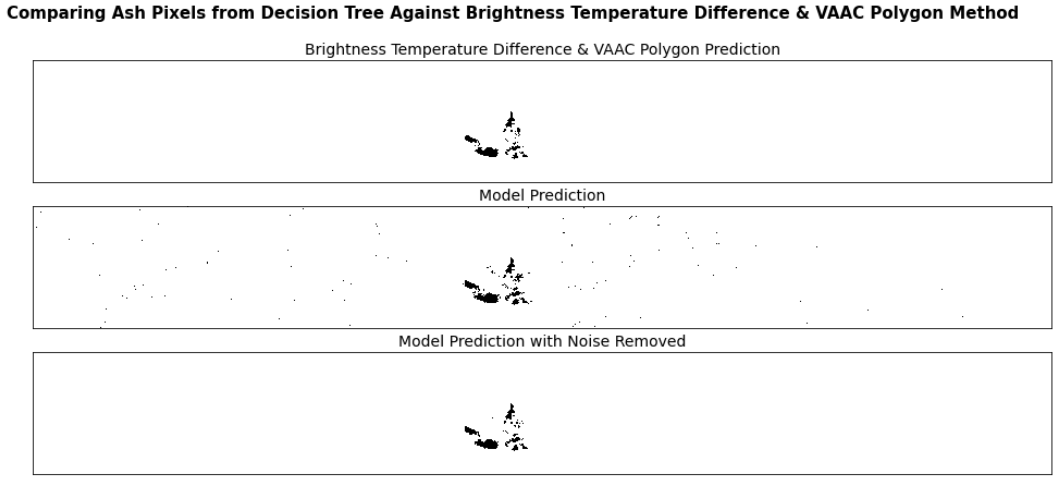


Figure 3.3: Decision tree model prediction for data set A.1.

Chapter 4

Conclusion

This project has investigated the suitability of using machine learning algorithms to determine the presence of volcanic ash clouds using data from the Himawari-8 satellite. In particular, support vector machine (SVM) models have been explored when trained on data sets created using an approach combining corrected brightness temperature readings produced from Himawari-8 data and polygons provided by the Tokyo Volcanic Ash Advisory Center for ash clouds emitted from Nishinoshima volcano during August and September of 2020.

Even when trained over the small scale of pixel subsets from just three scenes the SVM can be made to provide accurate predictions over a fourth scene and predicts minimal noise which can largely be removed in post-processing. This is despite the model being trained on potentially mislabeled pixels resulting from the use of a fixed-variable water vapour correction which has over-corrected the scenes in some cases.

Investigating the properties of the training data set found that the SVM performs best when providing it with a larger quantity of training data and training data sourced from a variety of scenes. While a large quantity of training data is beneficial, the SVM could be found to detect ash from a scene when tested on a very small subset of pixels from another. The largest issue was that the model could over-predict ash resulting in relatively large amounts of false positives. Unbalancing the training data set in favour of a higher quantity of 'not ash' data points proved beneficial by lowering the false positive count at the expense of a relatively small change in the number of false negative results. This could perhaps be explained by

there being a wider variety of scenarios encapsulated in the category of 'not ash', and therefore a larger quantity of data would be required to train the algorithm over this larger category subset. Similarly, training over a variety of scenes is beneficial because the training data encapsulates a wider range of these conditions resulting in a more robust machine learning model.

The type of SVM was also found to be important. In particular the kernel of the model significantly affected results, with the linear kernel providing models which generalised far better than non-linear kernel models as well as being faster to train. This better performance is likely because of the non-linear kernel models over-fitting the training data because those models performed well over scenes which had a subset of pixels involved in the model training. It is possible that investigating how the results vary as the hyperparameters of the models are changed would have resulted in an improvement of these results so this could be attempted in the future.

The linear kernel models were found to have coefficients which emphasised the importance of the 8.60 µm, 9.63 µm and 12.35 µm Himawari-8 channels in determining the classifying pixels as 'not ash', while the 11.20 µm and 10.45 µm channels indicated the presence of ash. Surprisingly, the shortwave channels were not found be very important for the detection of ash, despite ash appearing brown in the true colour image against blue water and white clouds. This may be worth investigating in the future, such as by removing the infrared channels from the model input to see how the algorithms perform on a scene.

A linear kernel SVM model trained on data from the Nishinoshima eruption was tested on an eruption of the volcano Raikoke, also located in the Pacific ocean. This model produced predictions which detected the ash cloud consistent to what can be seen in a true colour image of the scene. This suggests the model is robust enough to be extended to further locations, in this case without expanding the data set. A more detailed investigation could be carried out to find the full extent of the models generalisation abilities on other eruptions, which is left as work to be carried out in the future.

Decision tree and k-nearest neighbour classifiers were briefly investigated. Decision tree models were found to work on similar timescales to the SVM models with the training data used and produced results of a similar quality demonstrating the viability of this model for further study towards being used for ash classification. The k-nearest neighbour models were found to train quickly but take so long to run that images of the model predictions could not be made for this report. This extreme run time suggests that these types of models are unlikely to form the basis of successful ash classification algorithms, especially when the usefulness of SVMs and decision trees has been shown.

This project has highlighted areas where future work could be undertaken. The water vapour correction could be improved by varying the correcting variable on

a per-scene basis, which would avoid over-corrections and thus provide more accurate data sets for training and testing. The weights of data types could also be altered in unbalanced data sets, such as when training over relatively large amounts of 'not ash' data, which may result in algorithms with a lower rate of false negatives. Furthermore, models could be trained over a larger number of scenes with larger training data sets which may improve model performance on unseen scenes. The models considered in this project also considered pixels on a purely individual basis. Introducing an approach which combines individual pixel information with the information of the pixels surrounding it may help to reduce noise in the model predictions, reducing the amount of data lost in post-processing.

References

[1] T. Casadevall, "Volcanic ash and aviation safety; proceedings of the first international symposium on volcanic ash and aviation safety," 1994. [Cited on page 2]

[2] J.-P. e. a. Vernier, "In situ and space-based observations of the kelud volcanic plume: The persistence of ash in the lower stratosphere," *Journal of Geophysical Research: Atmospheres, Vol. 121, Issue 18*, 2016. [Cited on page 2]

[3] Y. e. a. Zhu, "Persisting volcanic ash particles impact stratospheric so2 lifetime and aerosol optical properties," *Nature Communications, Vol. 11*, 2020. [Cited on page 2]

[4] E. G. Dutton and J. R. Christy, "Solar radiative forcing at selected locations and evidence for global lower tropospheric cooling following the eruptions of el chichón and pinatubo," *Geophysical Research Letters, Vol. 19, Issue 23*, 1992. [Cited on page 2]

[5] A. Robock, "Volcanic eruptions and climate," *Reviews of Geophysics, Vol. 38, Issue 2*, 2000. [Cited on page 2]

[6] B. Langmann, "On the role of climate forcing by volcanic sulphate and volcanic ash," *Advances in Meteorology*, 2014. [Cited on page 2]

[7] T. Gray and R. Bennartz, "Automatic volcanic ash detection from modis observations using a back-propagation neural network," *Atmospheric Measurement Techniques Discussions*, vol. 8, pp. 8753–8777, 08 2015. [Cited on page 3]

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`. [Cited on pages 3 and 4]

[9] 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*, vol. 12, pp. 2825–2830, 2011. [Cited on pages 4 and 10]

[10] E. Venzke, "Volcanoes of the world - nishinoshima (284096)," 8 2021. Accessed 31 August 2021. [Cited on page 5]

[11] A. Voiland, "Raikoke erupts." Accessed 17/10/2021. [Cited on page 6]

[12] S. K. e. Sennert, "Weekly volcanic activity report, 26 june-2 july 2019." Accessed 17/10/2021. [Cited on page 6]

[13] "Tokyo volcanic ash advisory centre." [Cited on page 6]

[14] K. e. a. Bessho, "An introduction to himawari-8/9–japan's new-generation geostationary meteorological satellites, 94, 151–183," *Journal of the Meteorological Society of Japan*, 2016. [Cited on page 6]

[15] T. e. a. Yu, "Atmospheric correction for satellite-based volcanic ash mapping and retrievals using "split window" ir data from goes and avhrr," *Journal of Geophysical Research: Atmospheres*, 2002. [Cited on page 7]

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000. [Cited on page 11]

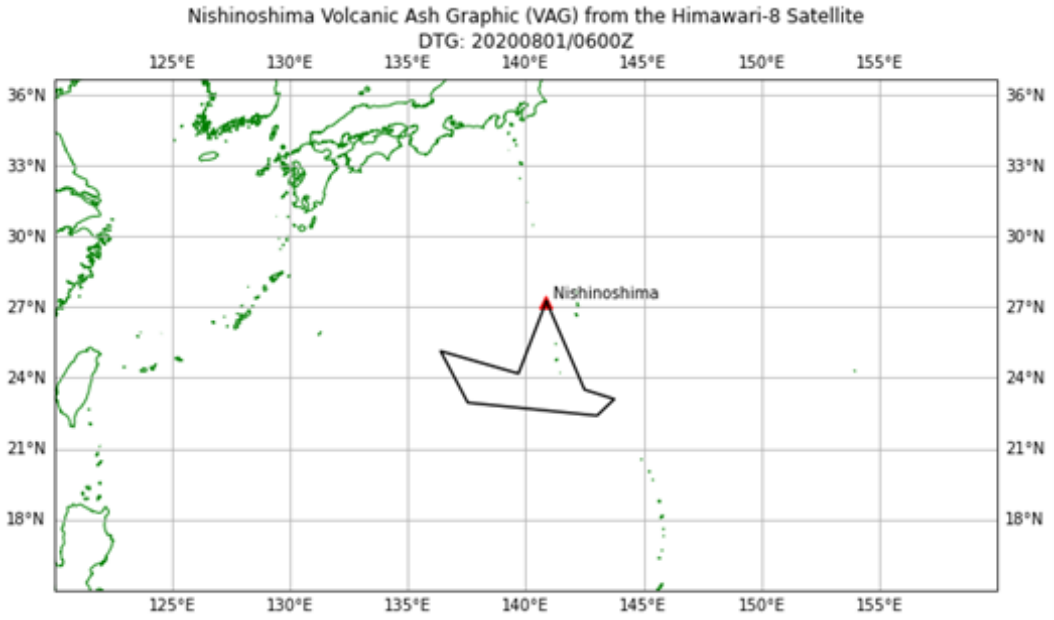
[17] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi, "scikit-optimize/scikit-optimize," Sept. 2020. [Cited on page 12]

Appendix A

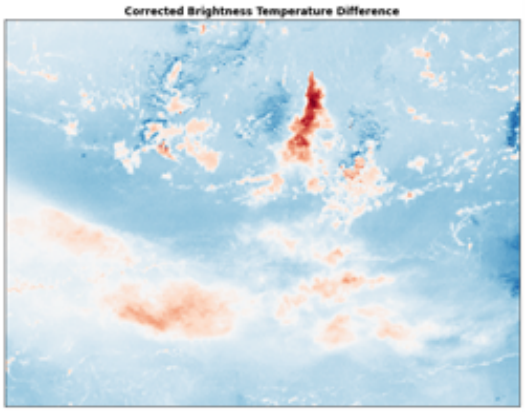
Data Sets

The data sets used in this project are represented in this appendix.

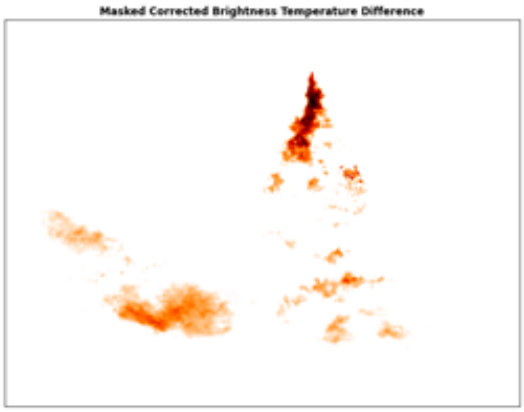
A.1 Nishinoshima Data Sets



(a) Volcanic Ash Graphic (VAG) showing the Tokyo VAAC polygon.

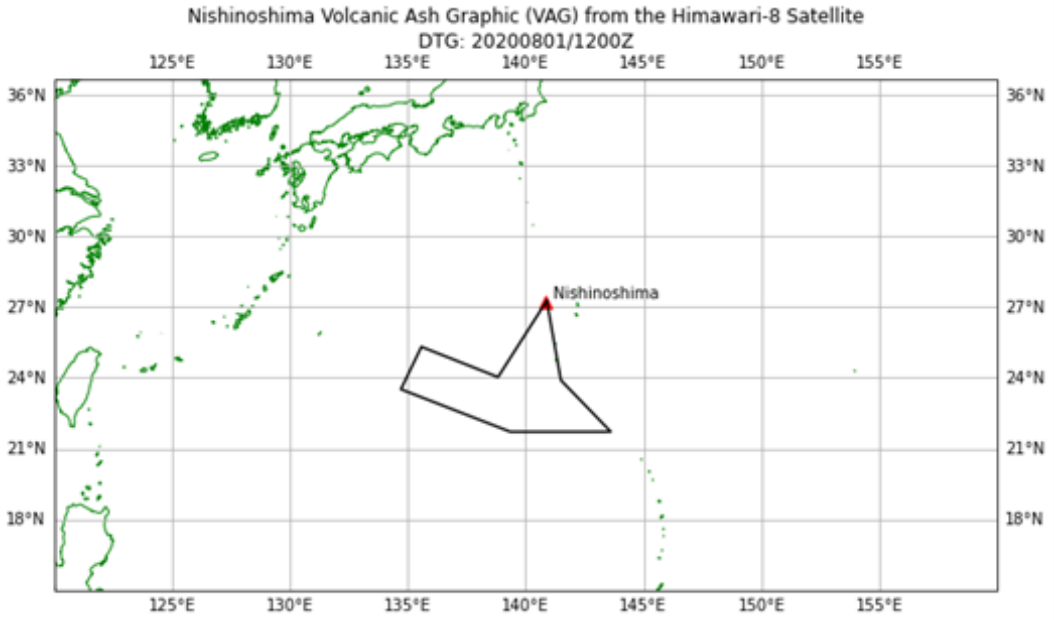


(b) Visualisation of the corrected brightness temperature (BTD) for each pixel, with red demonstrating the presence of ash and blue demonstrating a lack of ash.

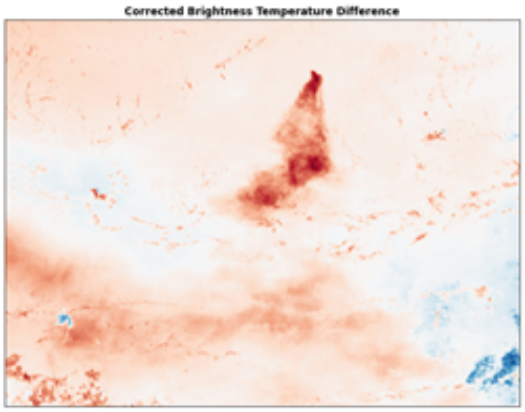


(c) Final data set created by combining the VAAC polygon and corrected BTD approaches.

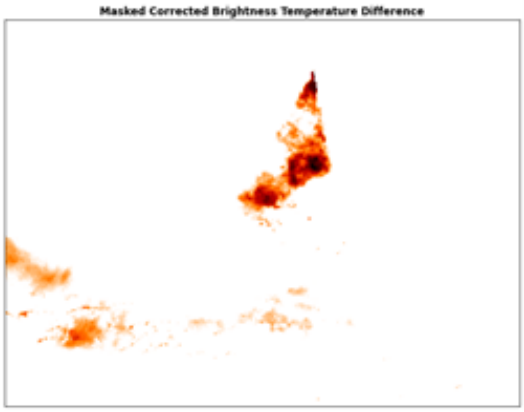
Figure A.1: Creating the dataset for 06:00 UTC on 01/08/2020



(a) Volcanic Ash Graphic (VAG) showing the Tokyo VAAC polygon.

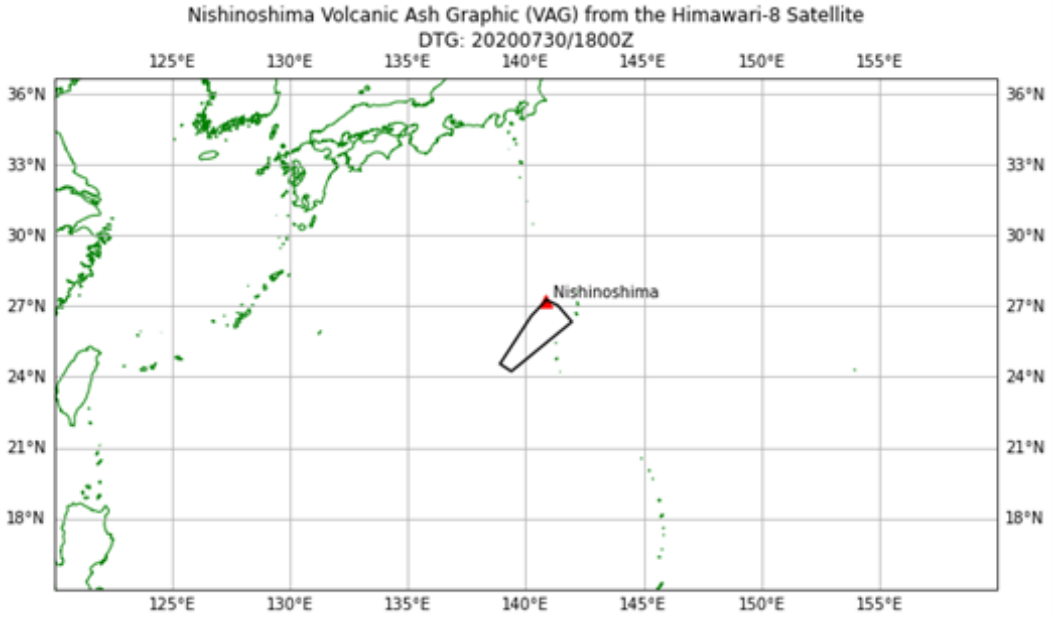


(b) Visualisation of the corrected brightness temperature (BTD) for each pixel, with red demonstrating the presence of ash and blue demonstrating a lack of ash.

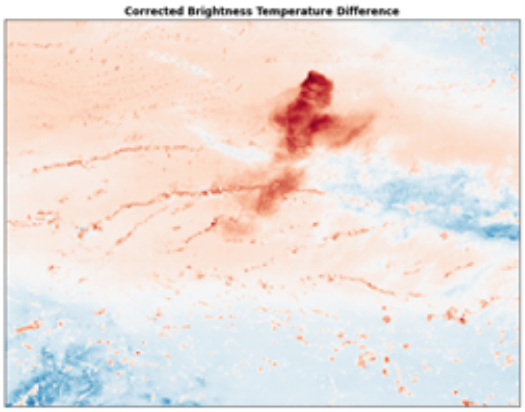


(c) Final data set created by combining the VAAC polygon and corrected BTD approaches.

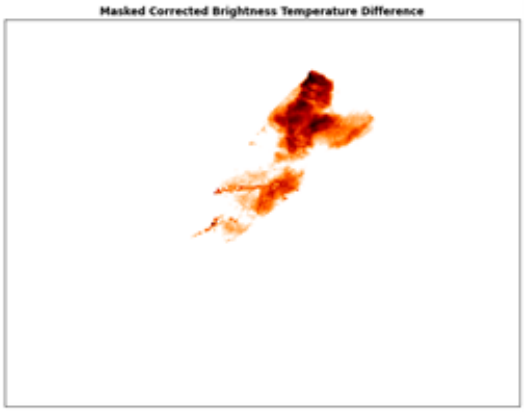
Figure A.2: Creating the dataset for 12:00 UTC on 01/08/2020



(a) Volcanic Ash Graphic (VAG) showing the Tokyo VAAC polygon.

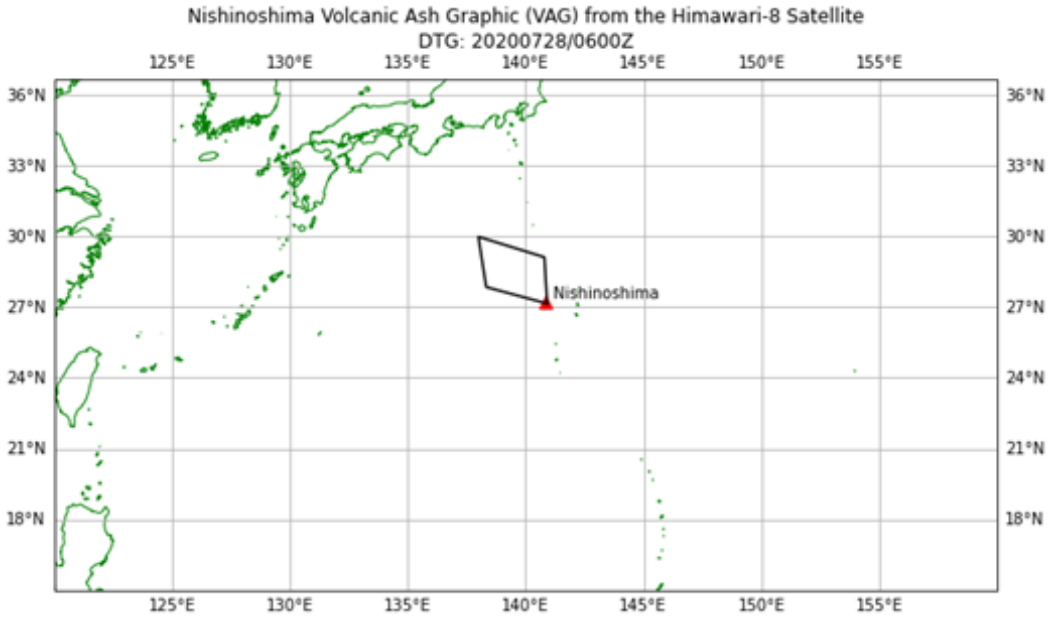


(b) Visualisation of the corrected brightness temperature (BTD) for each pixel, with red demonstrating the presence of ash and blue demonstrating a lack of ash.

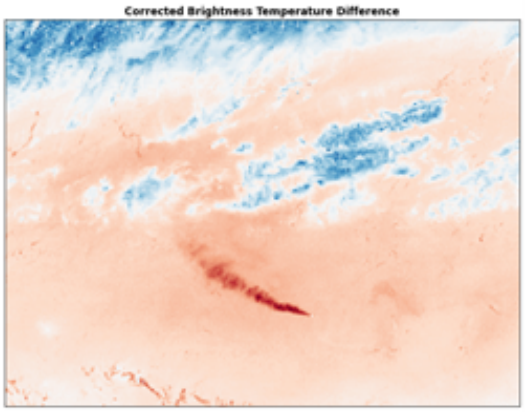


(c) Final data set created by combining the VAAC polygon and corrected BTD approaches.

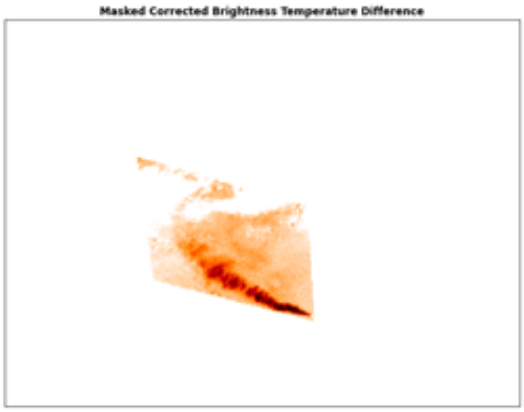
Figure A.3: Creating the dataset for 18:00 UTC on 30/07/2020



(a) Volcanic Ash Graphic (VAG) showing the Tokyo VAAC polygon.



(b) Visualisation of the corrected brightness temperature (BTD) for each pixel, with red demonstrating the presence of ash and blue demonstrating a lack of ash.



(c) Final data set created by combining the VAAC polygon and corrected BTD approaches.

Figure A.4: Creating the dataset for 16:00 UTC on 28/07/2020

Appendix B

Model Prediction Visualisations

B.1 Linear vs. Non-linear Kernel Visualisations

B.1.1 Linear Kernel Predictions

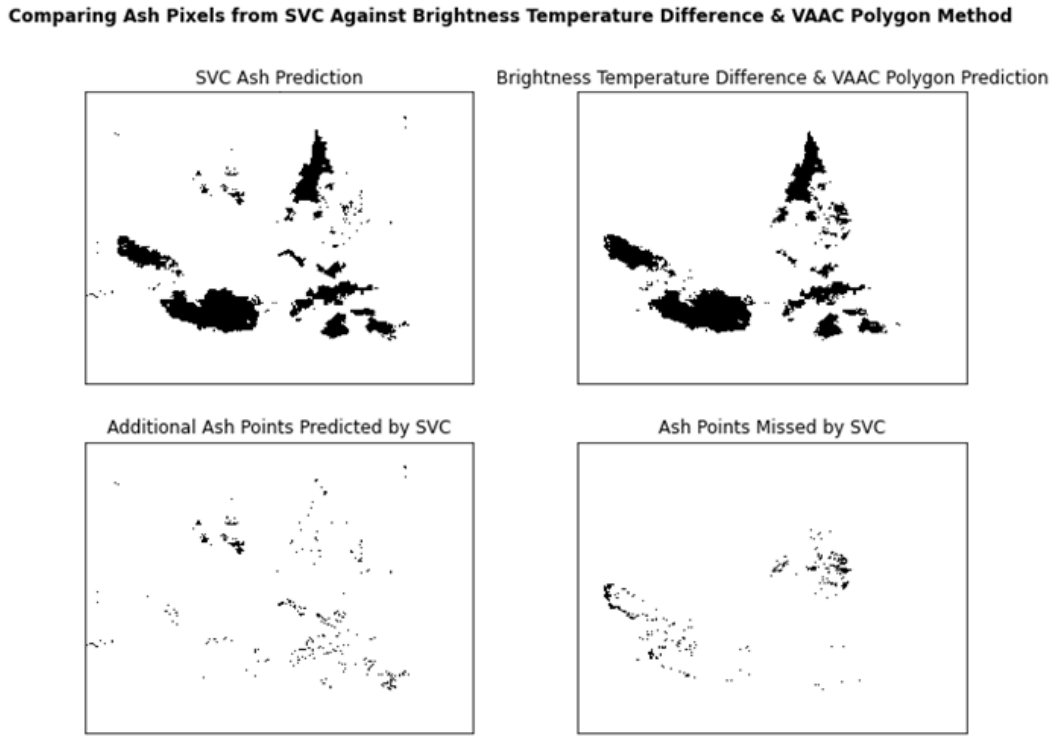


Figure B.1: Linear kernel model prediction for data set A.1.

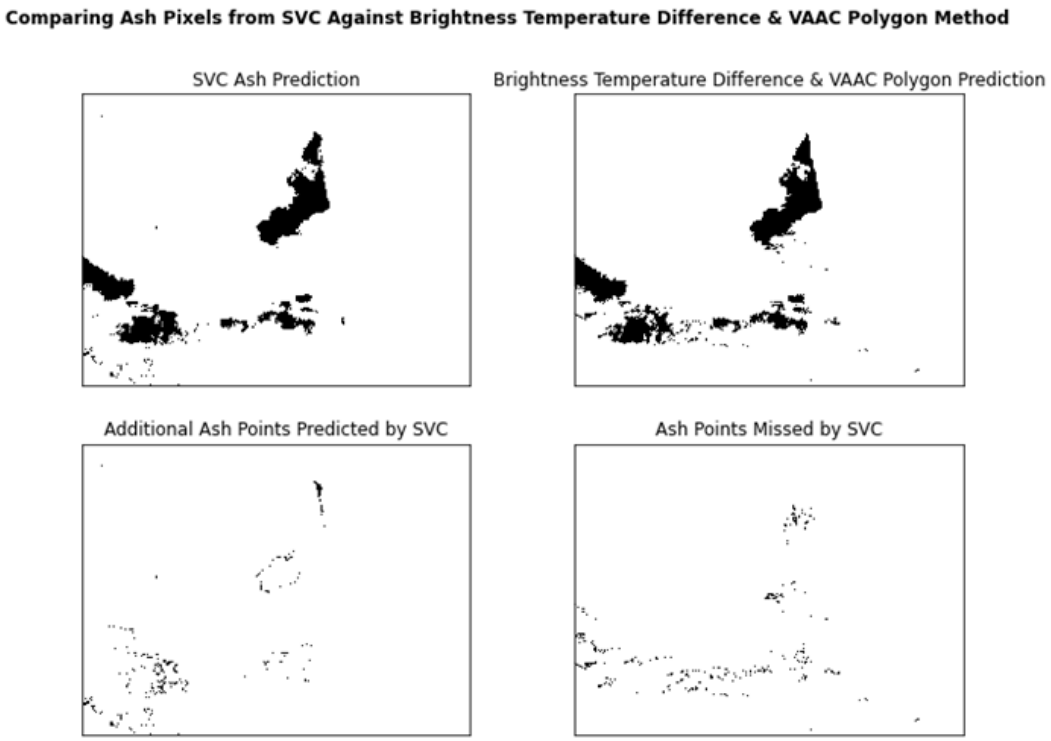


Figure B.2: Linear kernel model prediction for data set A.2.

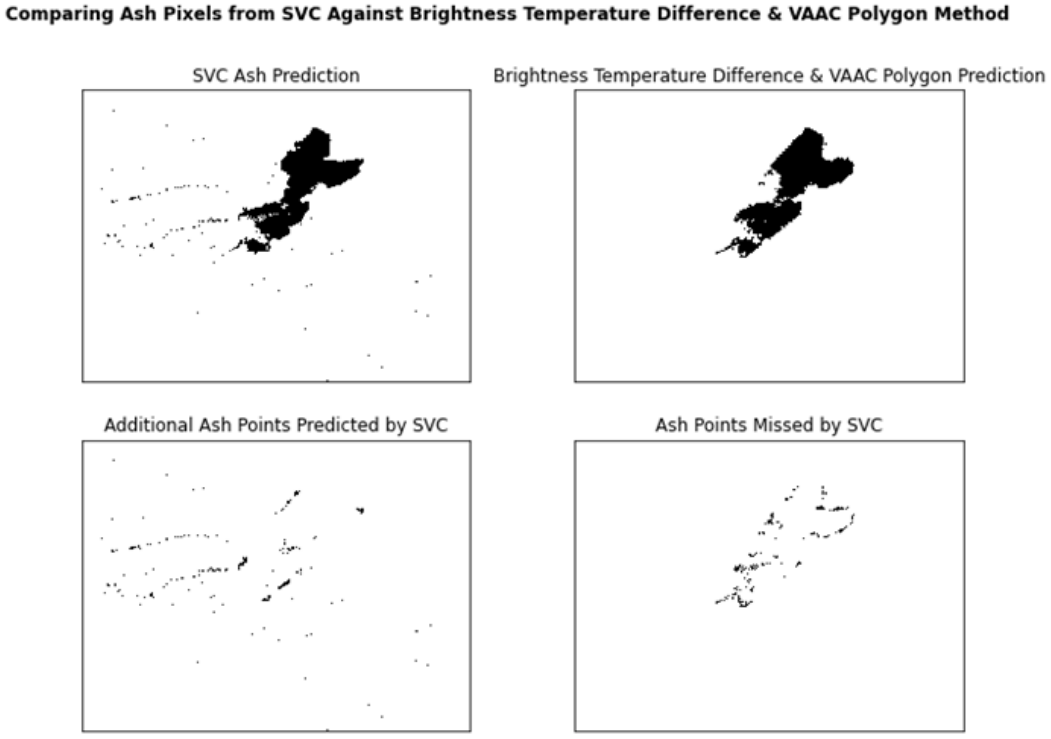


Figure B.3: Linear kernel model prediction for data set A.3.

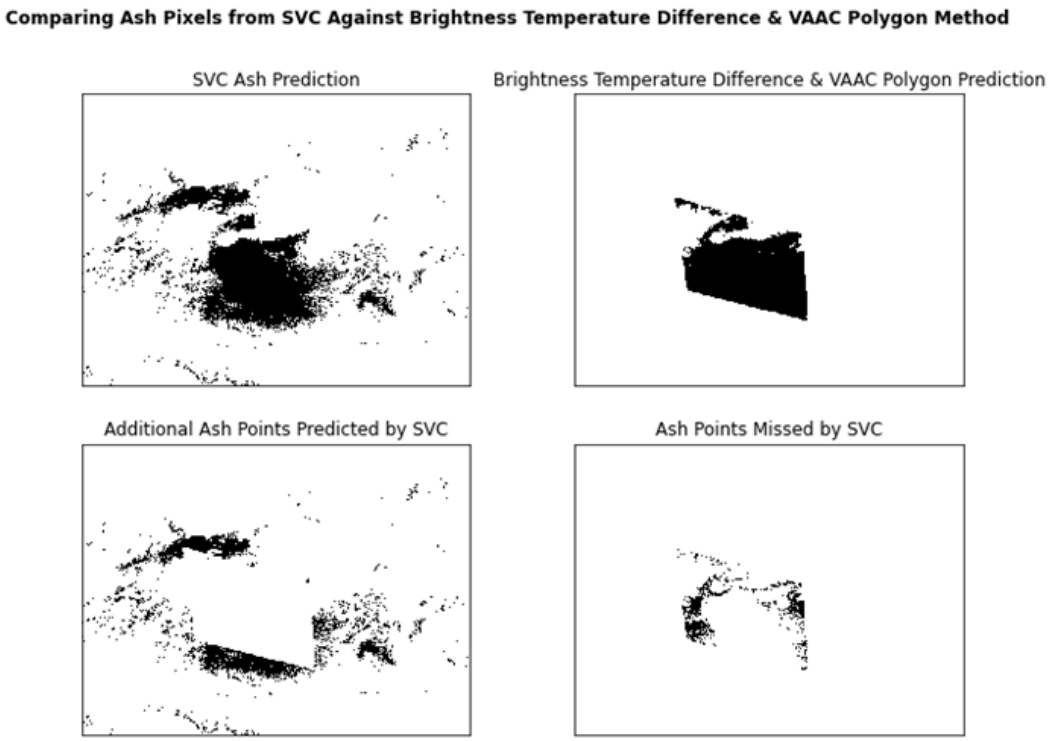


Figure B.4: Linear kernel model prediction for data set A.4.

B.1.2 Non-linear Kernel Predictions

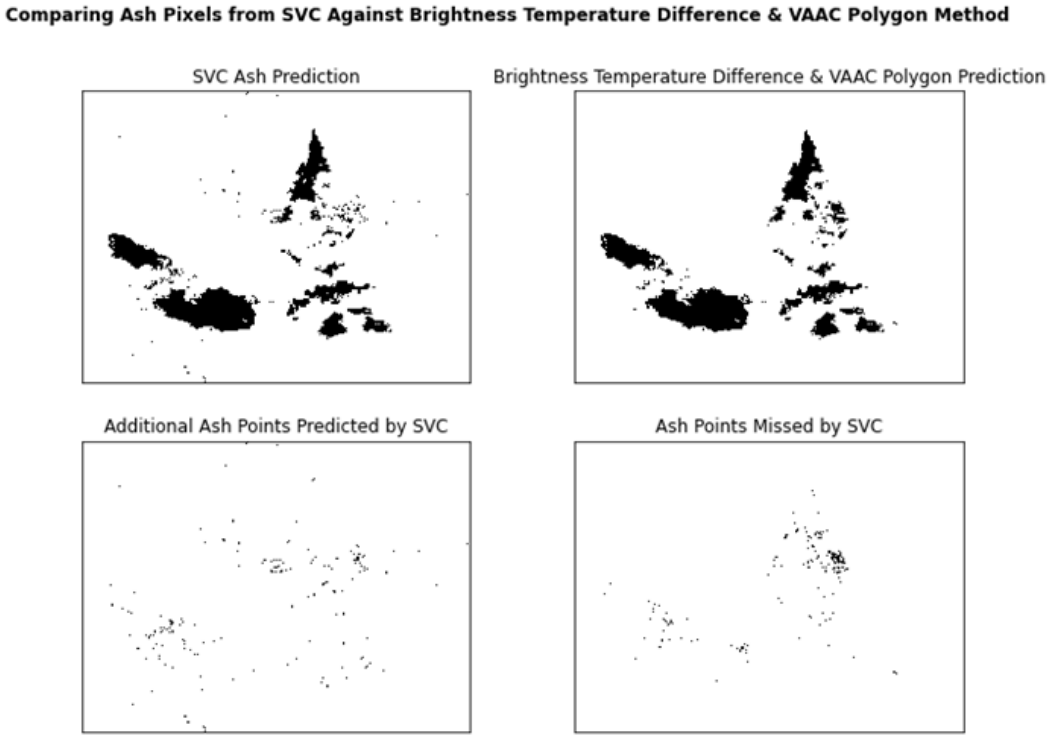


Figure B.5: Non-linear kernel model prediction for data set A.1.

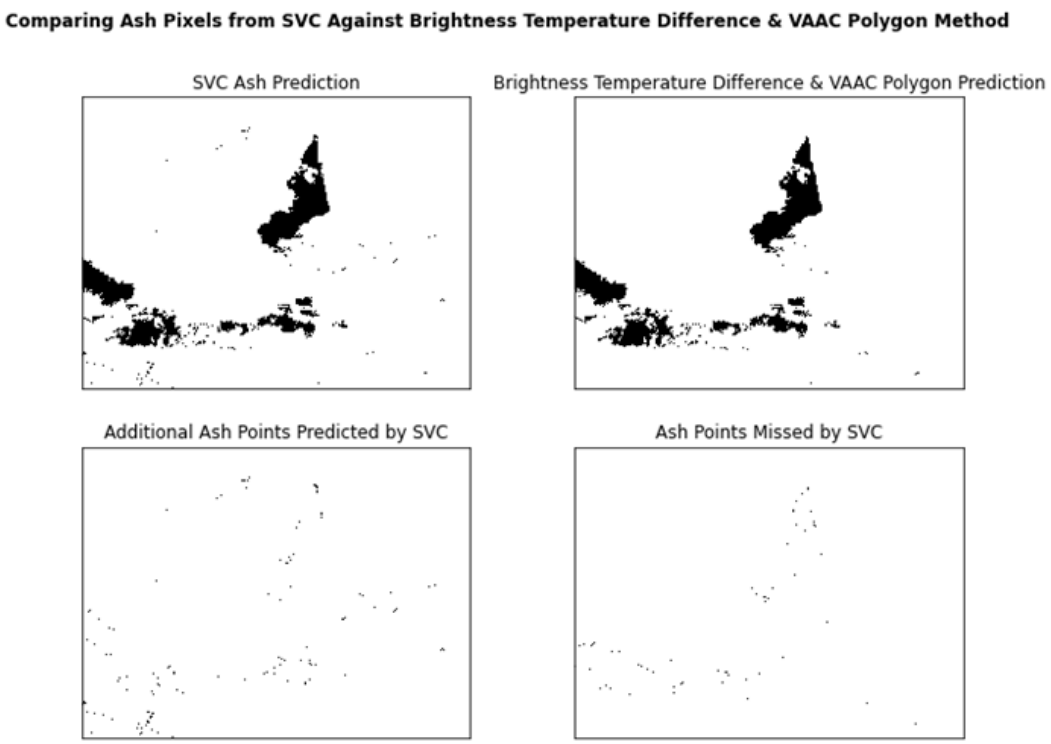


Figure B.6: Non-linear kernel model prediction for data set A.2.

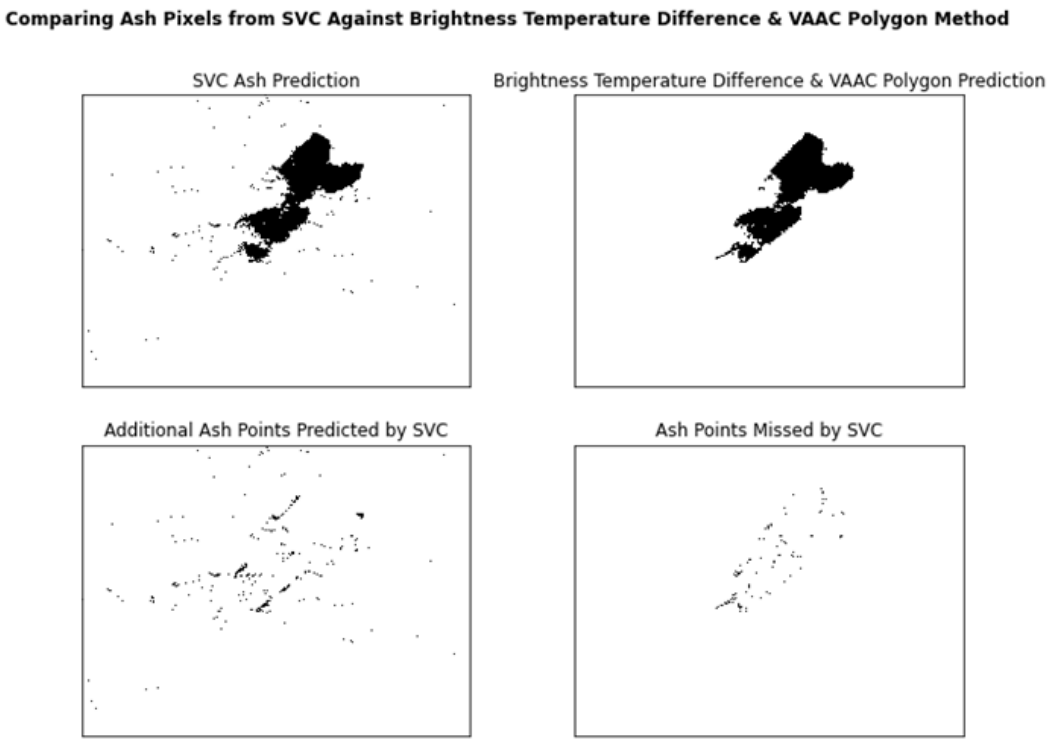


Figure B.7: Non-linear kernel model prediction for data set A.3.

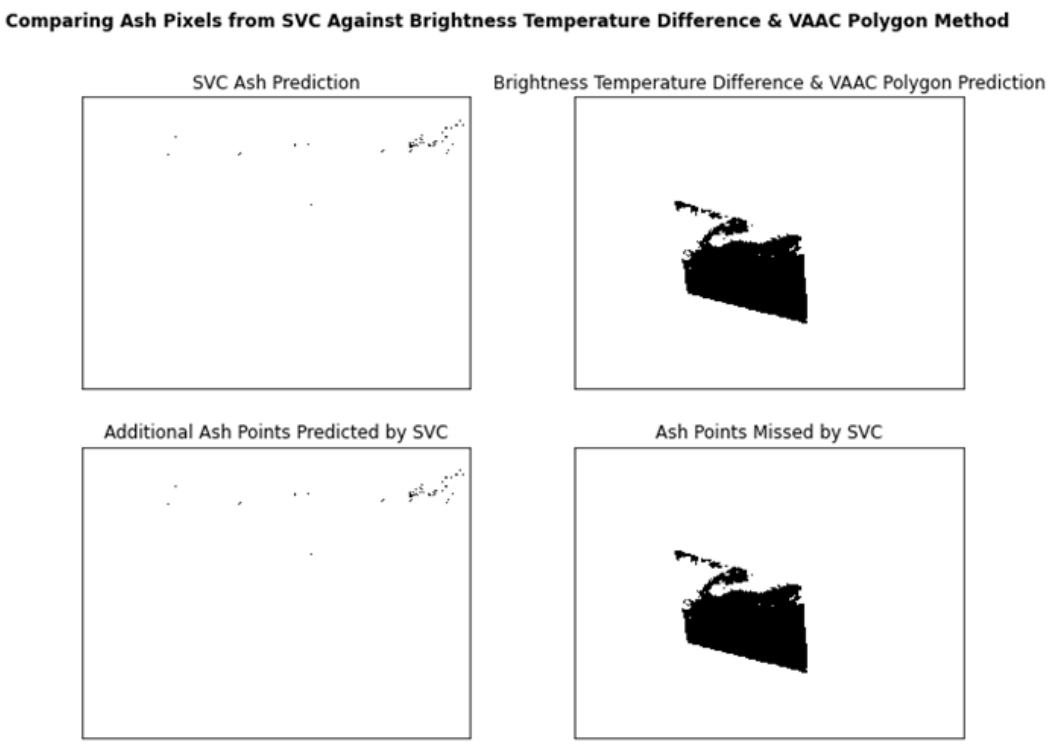


Figure B.8: Non-linear kernel model prediction for data set A.4.

B.2 Varying Quantity of Training Data Visualisations

B.2.1 More Training Data Predictions

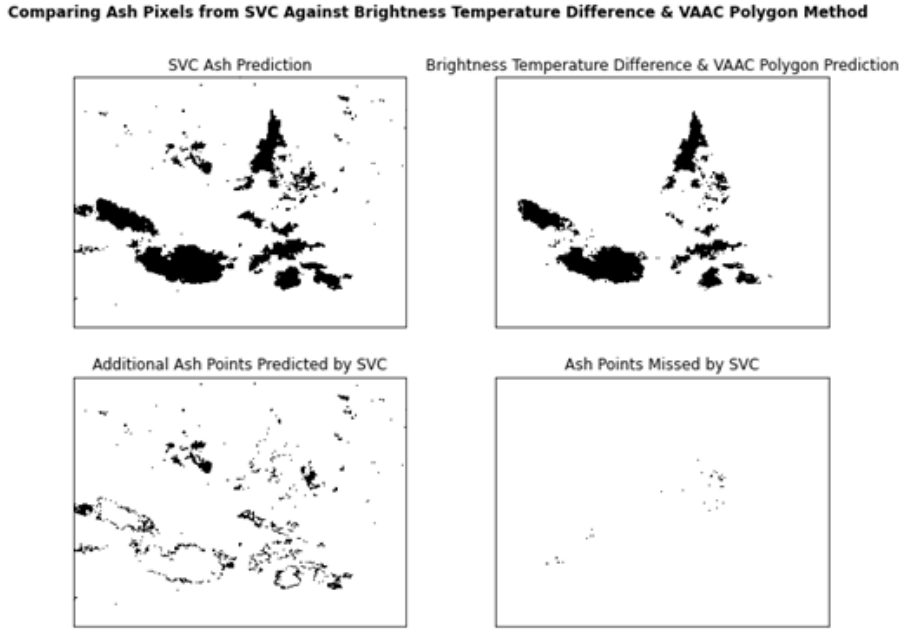


Figure B.9: Linear kernel model prediction for data set A.1 when trained on a higher quantity of data from data set A.1.

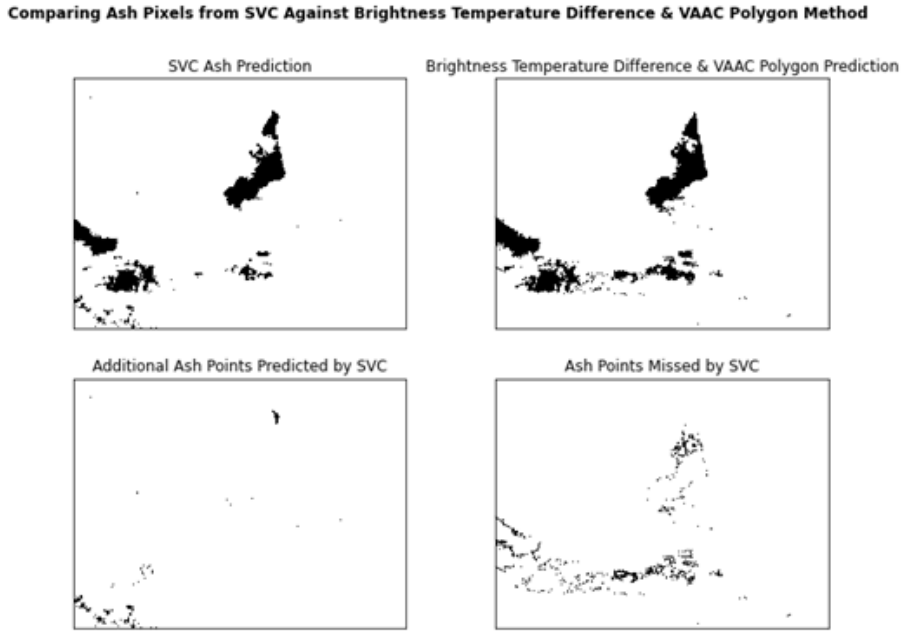


Figure B.10: Linear kernel model prediction for data set A.2 when trained on a higher quantity of data from data set A.1.

B.2.2 Less Training Data Predictions

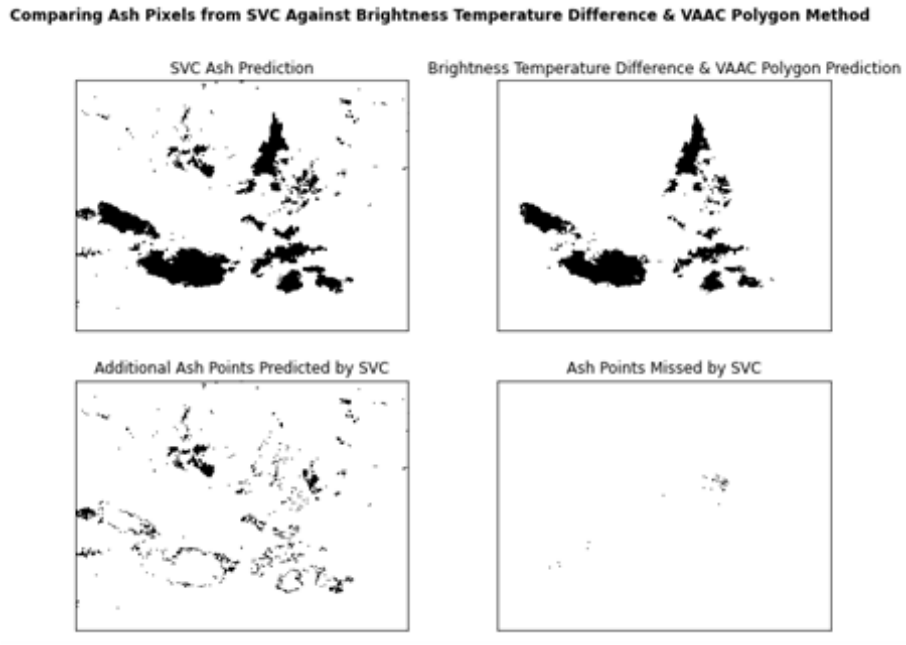


Figure B.11: Linear kernel model prediction for data set A.1 when trained on a lower quantity of data from data set A.1.

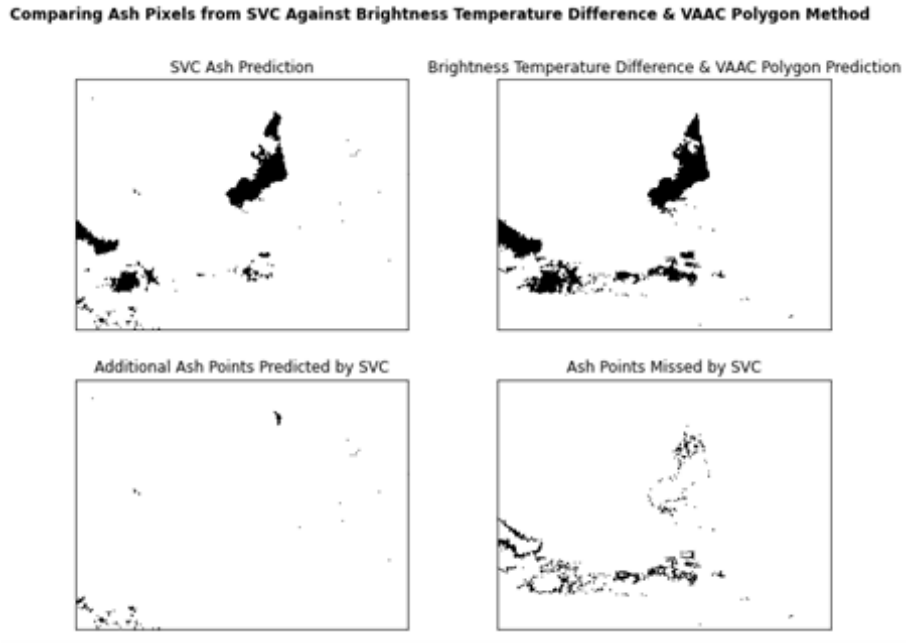


Figure B.12: Linear kernel model prediction for data set A.2 when trained on a lower quantity of data from data set A.1.

B.2.3 Unbalanced Training Data Predictions

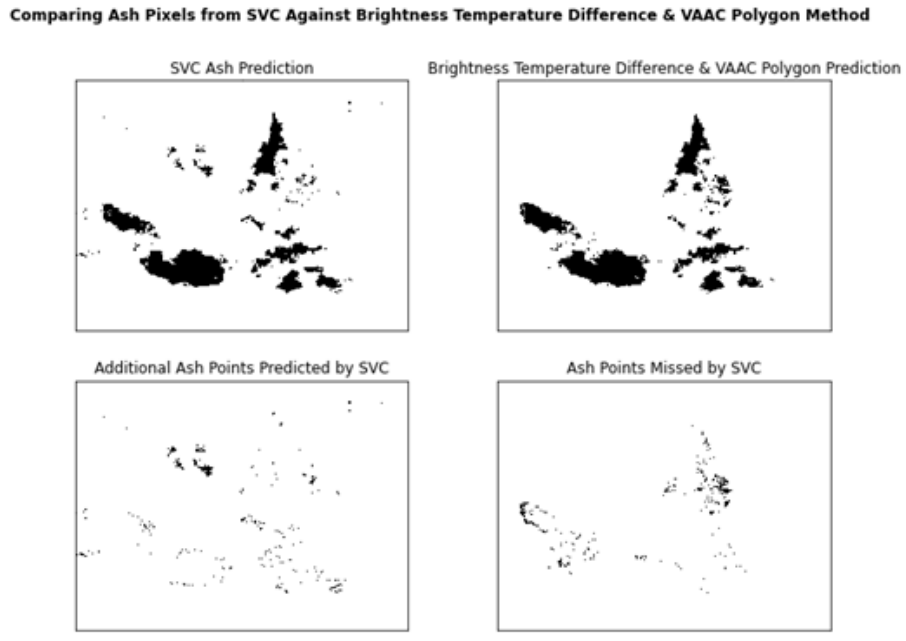


Figure B.13: Linear kernel model prediction for data set A.1 when trained on a higher ratio of 'not ash' data points to 'ash' data points from data set A.1.

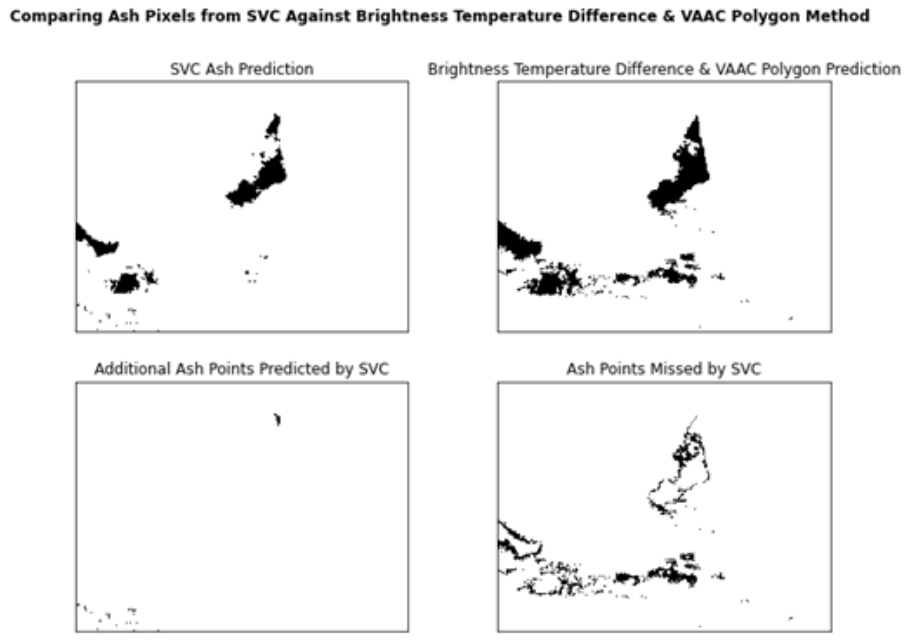


Figure B.14: Linear kernel model prediction for data set A.2 when trained on a higher ratio of 'not ash' data points to 'ash' data points from data set A.1.

B.3 Changing Variety of Training Data Visualisations

B.3.1 Algorithm Trained on Single Scene Predictions

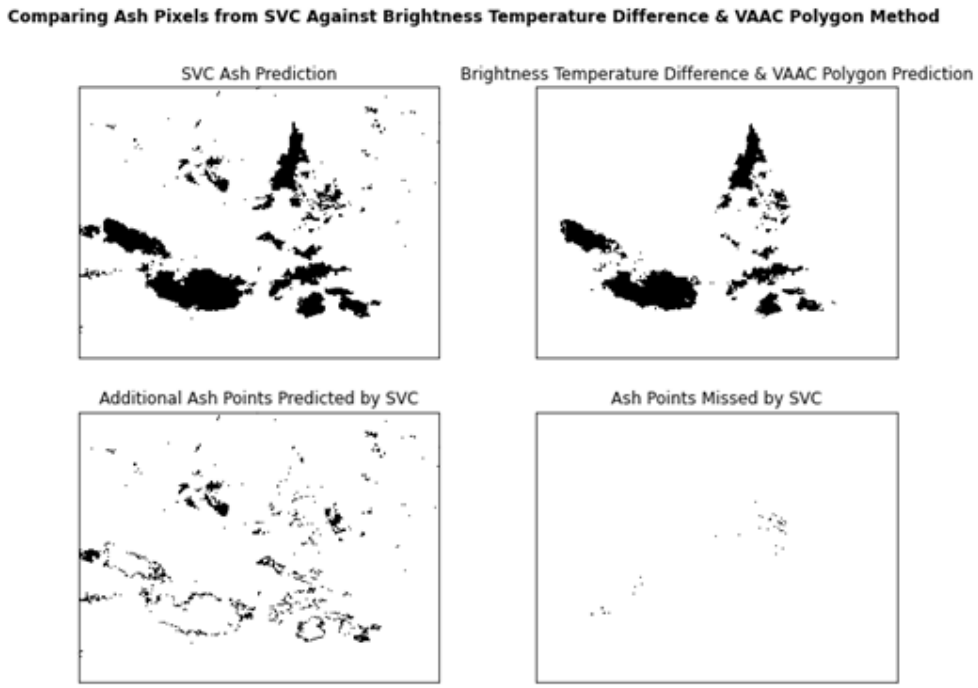


Figure B.15: Linear kernel model prediction for data set A.1 when trained on just data set A.1.

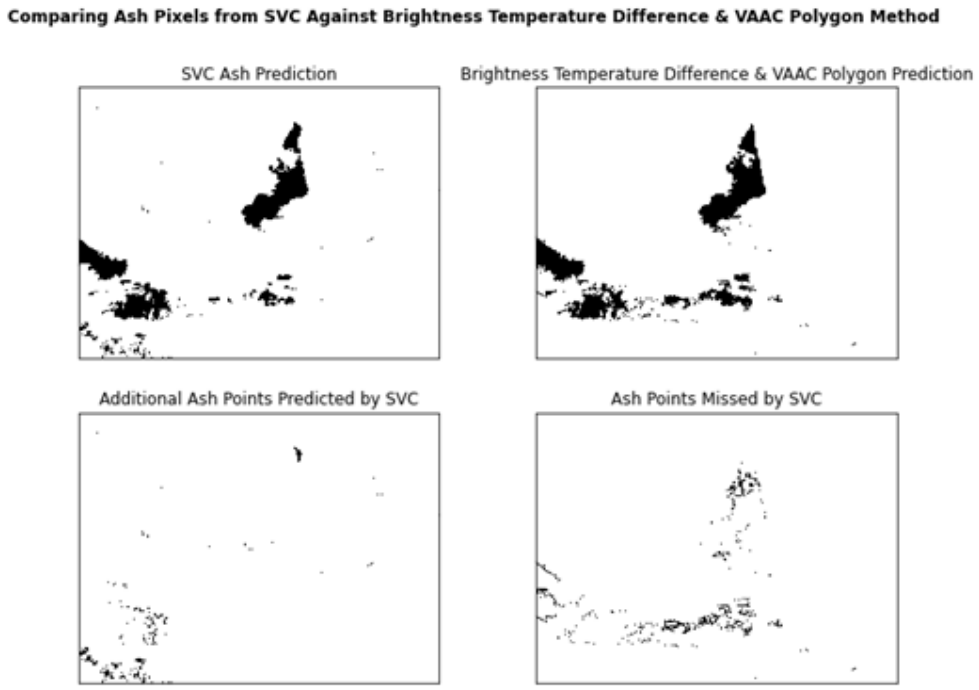


Figure B.16: Linear kernel model prediction for data set A.2 when trained on just data set A.1.

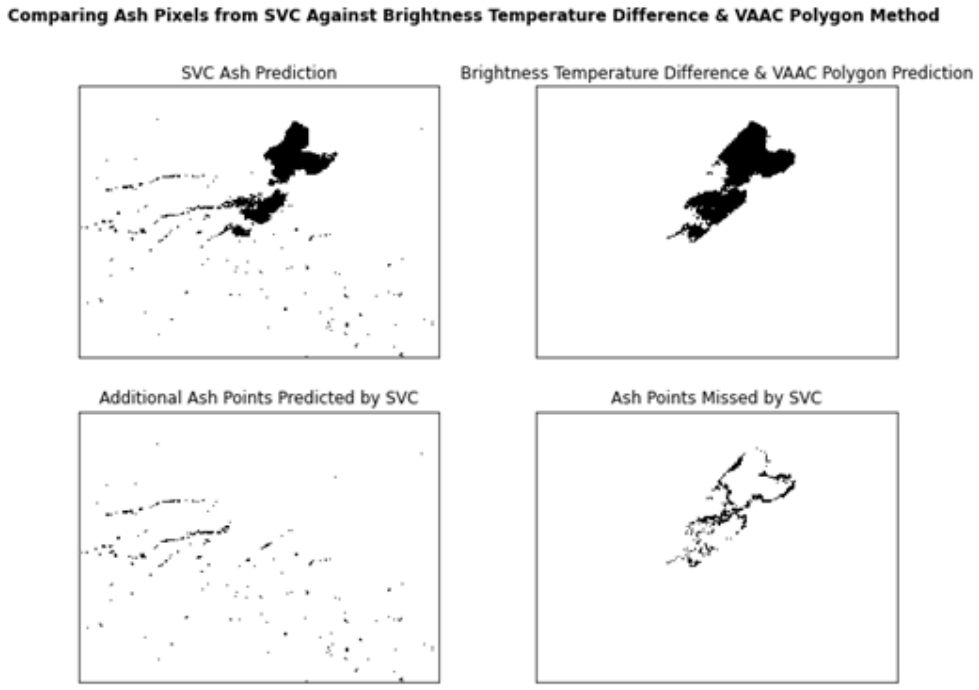


Figure B.17: Linear kernel model prediction for data set A.3 when trained on just data set A.1.

B.3.2 Algorithm Trained on Three Scenes Predictions

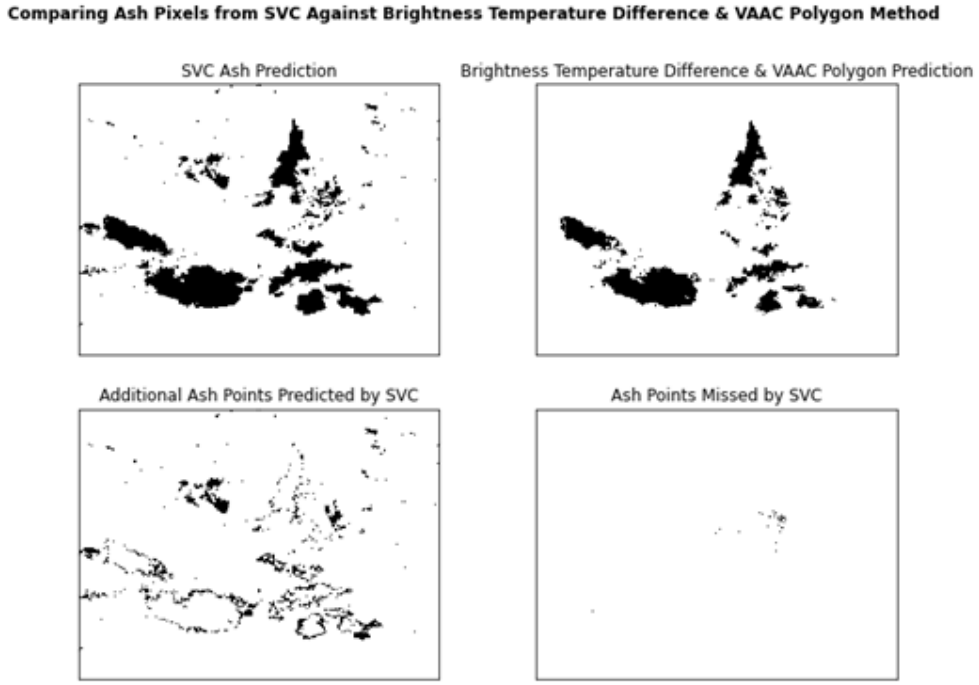


Figure B.18: Linear kernel model prediction for data set A.1 when trained on data sets A.1, A.2 and A.3.

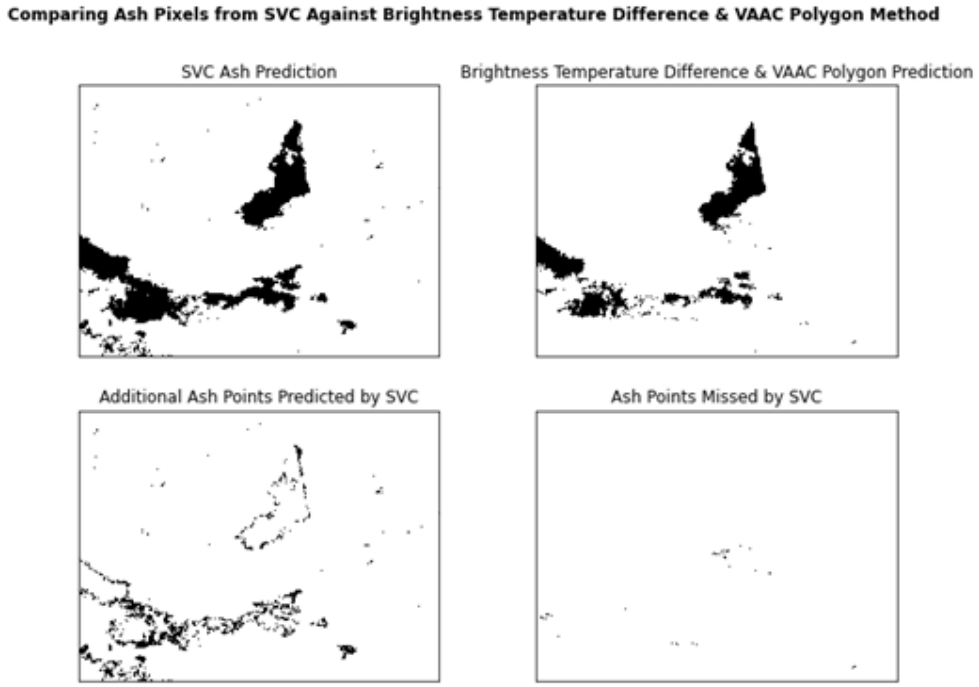


Figure B.19: Linear kernel model prediction for data set A.2 when trained on data sets A.1, A.2 and A.3.

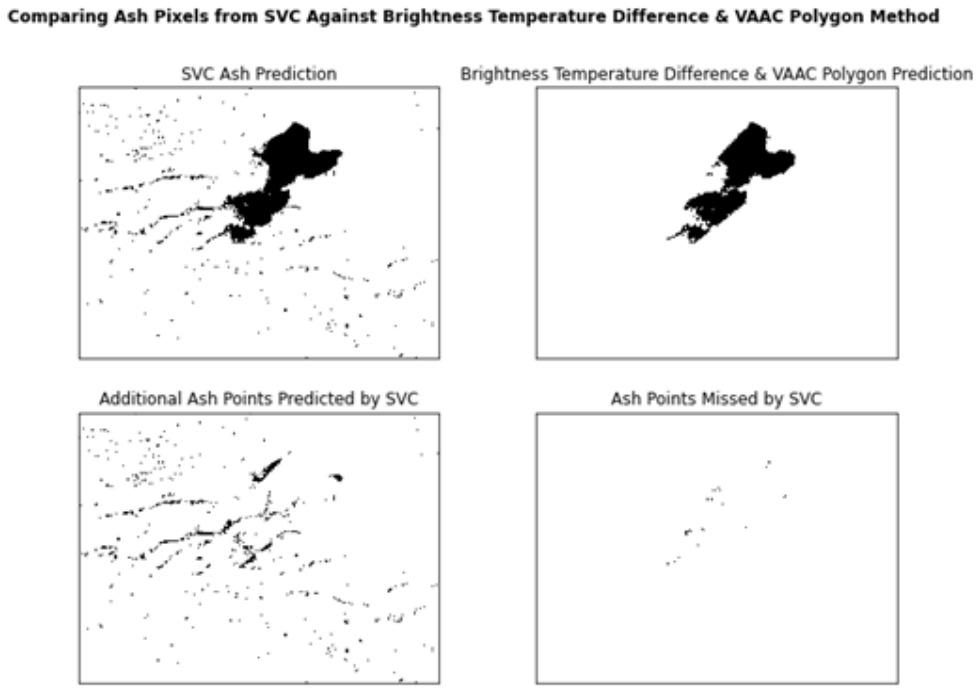


Figure B.20: Linear kernel model prediction for data set A.3 when trained on data sets A.1, A.2 and A.3.

B.4 Full Segment Model Predictions

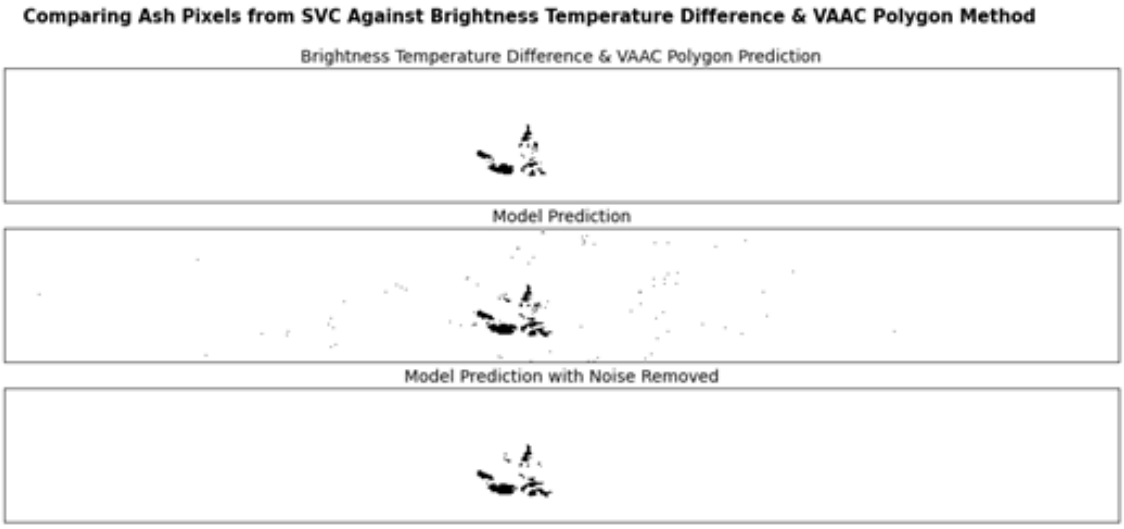


Figure B.21: Linear kernel model prediction for full Himawari-8 data set segment A.1 when trained on data sets A.1, A.2 and A.3 with twenty times as many 'not ash' data points as 'ash' data points.

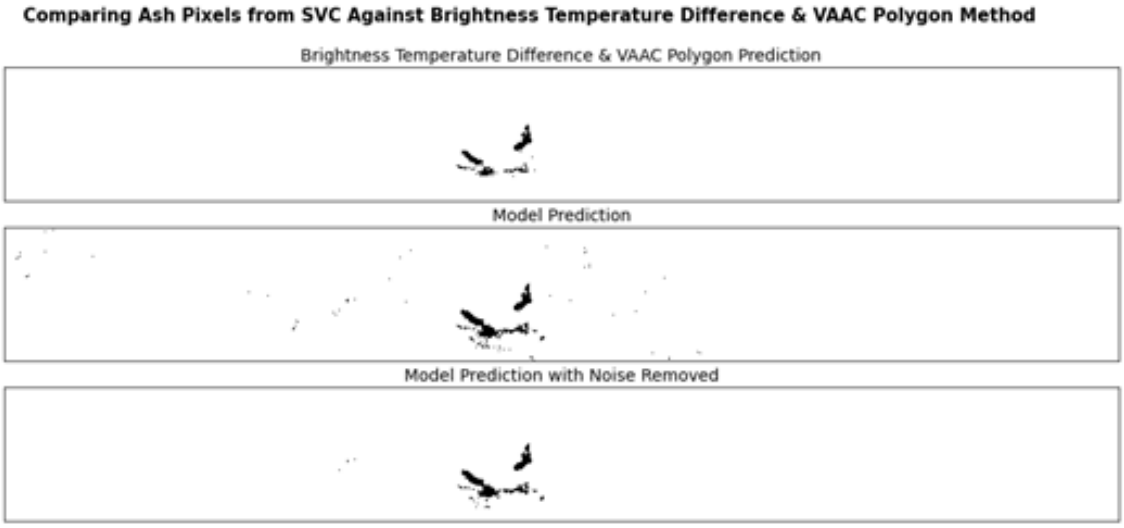


Figure B.22: Linear kernel model prediction for full Himawari-8 data set segment A.2 when trained on data sets A.1, A.2 and A.3 with twenty times as many 'not ash' data points as 'ash' data points.

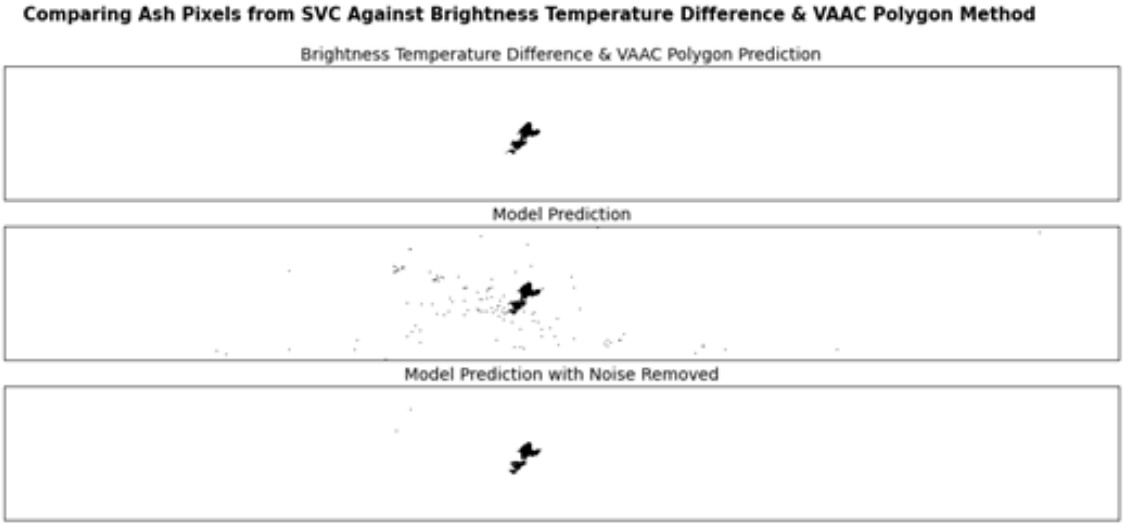


Figure B.23: Linear kernel model prediction for full Himawari-8 data set segment A.3 when trained on data sets A.1, A.2 and A.3 with twenty times as many 'not ash' data points as 'ash' data points.

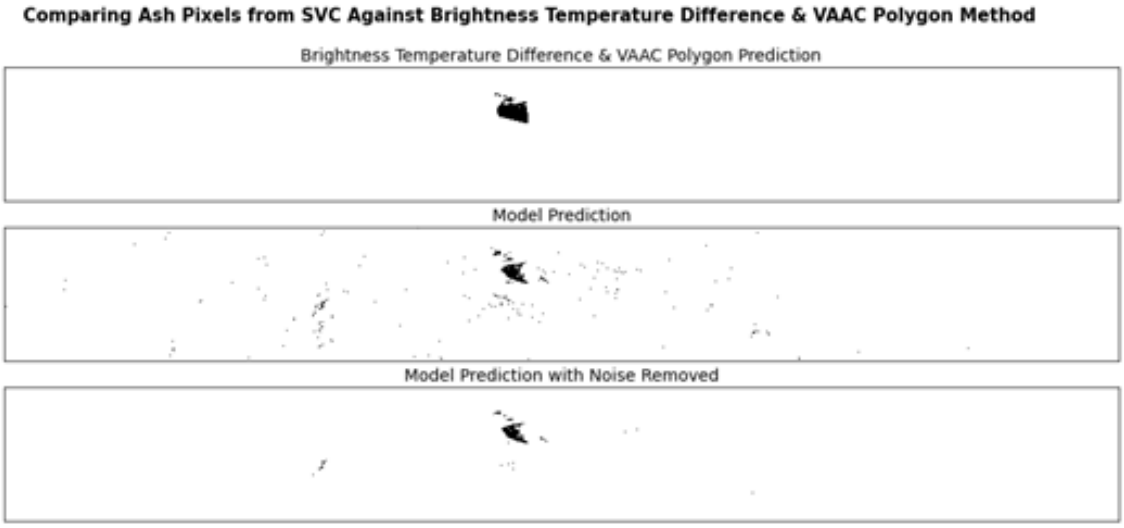


Figure B.24: Linear kernel model prediction for full Himawari-8 data set segment A.4 when trained on data sets A.1, A.2 and A.3 with twenty times as many 'not ash' data points as 'ash' data points.

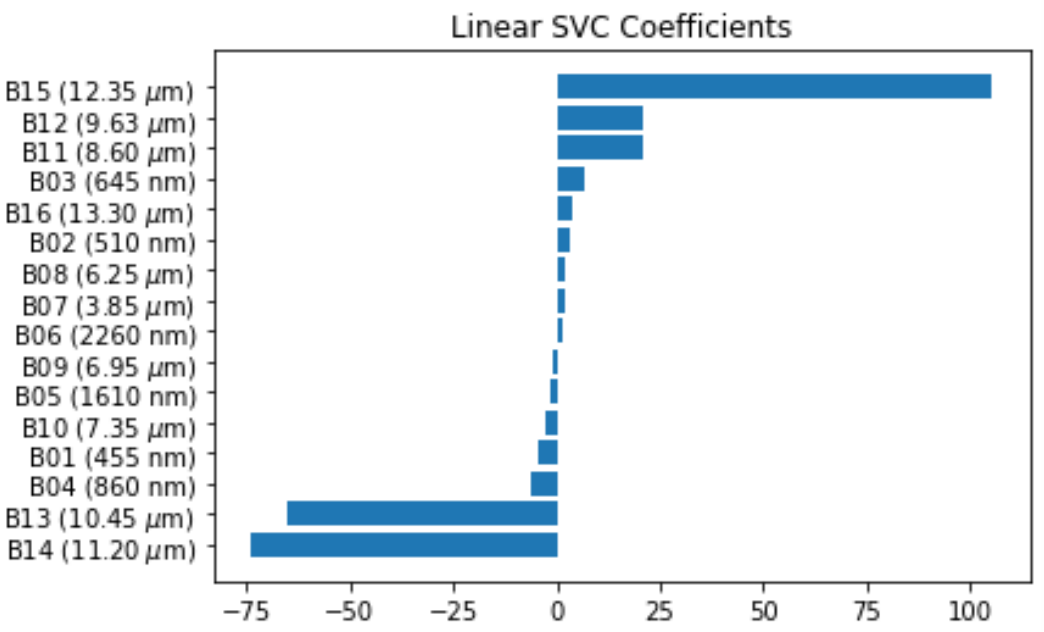


Figure B.25: Linear kernel model coefficients when trained on data sets A.1, A.2 and A.3 with twenty times as many 'not ash' data points as 'ash' data points.

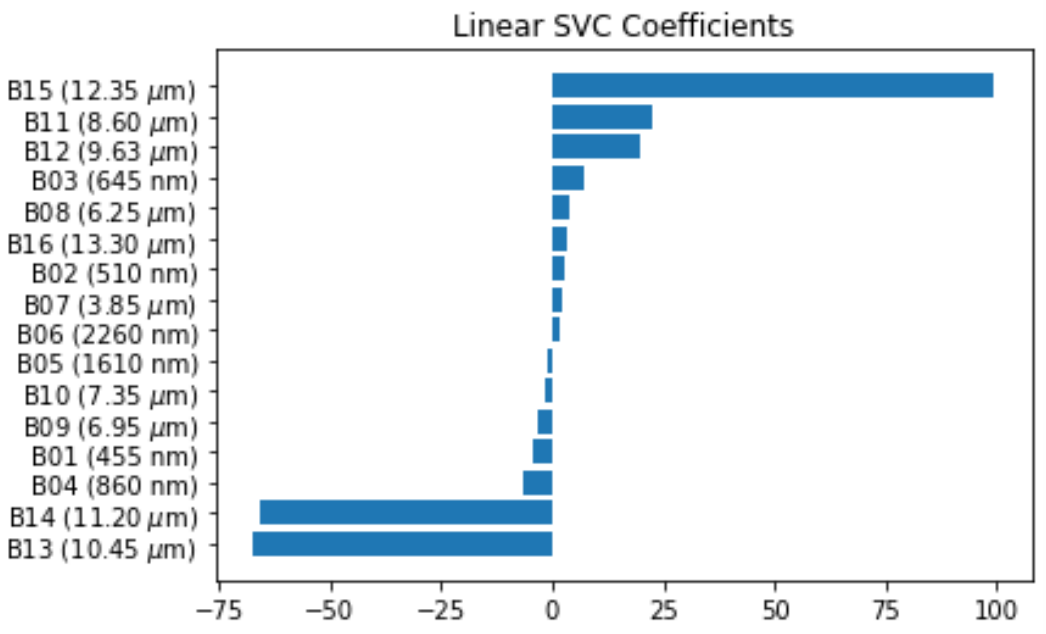


Figure B.26: Linear kernel model coefficients when re-trained on data sets A.1, A.2 and A.3 with twenty times as many 'not ash' data points as 'ash' data points. Notice there are small changes from B.25.