



Universiteit  
Leiden

# Master Computer Science

Automatic ship plume detection using a  
Convolutional Neural Network

Name: Jelle J. Lubben  
Student ID: 1288245  
Date: 30/06/2021  
Specialisation: Artificial intelligence  
1st supervisor: Dr. Cor J. Veenman  
2nd supervisor: Prof.dr.ir. Fons J. Verbeek  
Daily supervisor: Solomiia Kurchaba MSc

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)  
Leiden University  
Niels Bohrweg 1  
2333 CA Leiden  
The Netherlands

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Method</b>	<b>4</b>
2.1	CNN . . . . .	4
2.1.1	Convolutional layer . . . . .	4
2.1.2	Pooling . . . . .	5
2.1.3	Final layer and Activation functions . . . . .	5
2.1.4	Overfitting . . . . .	6
2.1.5	Network training . . . . .	7
2.2	Models . . . . .	7
2.3	Required data . . . . .	9
2.3.1	TROPOMI/Sentinel-5 . . . . .	9
2.3.2	AIS . . . . .	9
2.4	Regridding . . . . .	10
2.5	Pre-processing . . . . .	10
2.5.1	Mean . . . . .	10
2.5.2	Median . . . . .	11
2.5.3	Gaussian . . . . .	11
2.5.4	Bilateral . . . . .	12
2.5.5	Moran's I . . . . .	13
<b>3</b>	<b>Data</b>	<b>14</b>
3.1	Data selection . . . . .	15
<b>4</b>	<b>Classifier</b>	<b>16</b>
4.1	Results . . . . .	17
<b>5</b>	<b>Discussion</b>	<b>18</b>
<b>6</b>	<b>Conclusion</b>	<b>19</b>
<b>7</b>	<b>Future research</b>	<b>19</b>
<b>A</b>	<b>Appendix</b>	<b>22</b>
A.1	An example of the test set labelled "plume" . . . . .	22
A.2	An example of the test set labelled "no plume" . . . . .	28

## Abstract

In this paper we check whether NO<sub>x</sub> plumes produced by ships in the Mediterranean Sea that can be measured by satellite sensor can be recognized by a CNN in an automated way. Finding highly polluting ships can be important in the reduction of NO<sub>x</sub> emissions. We generate and label a dataset needed for this problem. We also explore how different image filtering techniques influence the performance of the CNN model. A lot of images generated for the dataset were not usable due to missing data or NO<sub>x</sub> produced on land. However, with the images that were clean enough for the dataset, an accuracy of 95.99% was reached with no pre-processing filters applied to the data.

# 1 Introduction

As of January 2020, new regulations on marine fuel has been enacted by the International Maritime Organization (IMO). These new regulations restrict the  $\text{NO}_x/\text{SO}_2$  presence in shipping fuel[1][2], what puts additional financial burden[3] on shipping companies thus given them incentive not to comply.

Due to the fact that the ships are moving on open sea, the measuring of the sulphur and nitrogen levels is difficult. Current methods for measuring the nitrogen emission for ships include taking fuel samples, checking the engine room logs and on-board measurements of the exhaust pipes of the ships[4]. Other more advanced methods involve airborne platforms[5], mobile platforms[6], measurement of reflected skylight for remote optical sensing with the DOAS technique[7], as well as other more advanced techniques such as [8],[9],[10],[11]. These methods are only feasible if you are in close proximity to the ships, thus can be performed only in ports or off the coast areas. The amount of ships coming and going through the ports however is too large for the local port authorities to fully check.

A new potential solution has been introduced by identifying distinct nitrogen plumes using TROMOPI/S5P satellite data. This instrument can capture data with a higher resolution than any previous satellite[12], thus can be used for identification of plumes produced by individual ships. In [12] it is shown that under certain conditions nitrogen plumes can be seen and linked to individual ships on the open sea.

In this paper we use Convolutional Neural Network(CNN), a model design for analyzing image data, to check in an automated way whether a plume produced by a ship is distinguishable by satellite sensor. We also study how the application of different image enhancement filters influences plume distinguishability. By means of AIS data we link plumes to a ship.

This thesis is a part of the project “Algorithms for the Verification of Emissions from Shipping with Satellites” (AVES-oculuS) for the Inspectorate for Human Environment and Transport (ILT) of the Netherlands.

This work is organized as follows: In Section 2.1 the structure of a CNN is explained. Section 2.2 is used for the introduction of the CNN architectures that are later used for the experiments. In Section 2.3 we present data necessary for the performance of this study. The data selection process is covered in Section 3. Finally, the results and discussions can be found in Section 4 and 5 respectively.

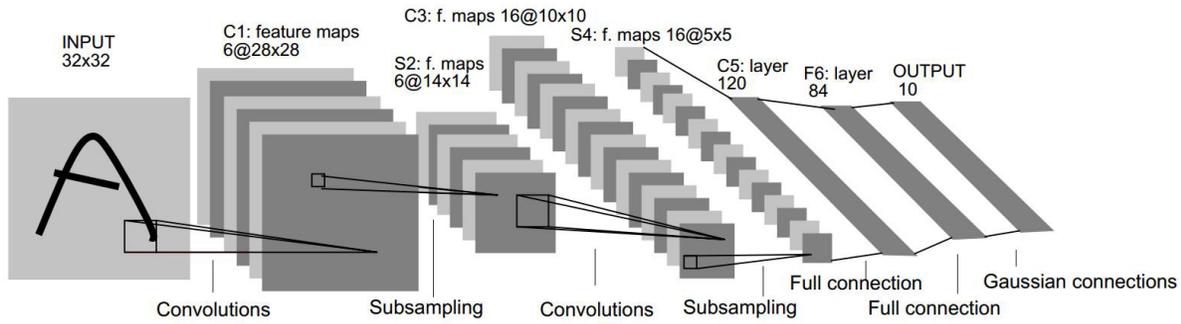


Figure 1. The architecture of the LeNet-5, a Convolutional Neural network(CNN) for classifying handwritten numbers. Each plane represents a feature map[14].

## 2 Method

### 2.1 CNN

CNN's are a subclass of deep neural networks most commonly applied for visual analysis. The LeNet architecture was introduced by [13] in 1989 as one of the earliest known CNN with the purpose of classifying handwritten numbers. This network was later improved to LeNet-5[14] so that the current version is able to classify handwritten numbers from the MNIST-dataset with an accuracy of 99.2%. The MNIST-dataset is a commonly used dataset for image processing techniques and consists of 60.000 training images and 10.000 testing images. This dataset is often used for comparing the effectiveness of different image processing techniques. The architecture of the LeNet-5 network is shown in Figure 1.

#### 2.1.1 Convolutional layer

In a classical multi-layered perceptron (MLP) the input is fully connected to the first hidden layer. When using a small RGB-image of  $32 \times 32 \times 3$  pixels for every hidden node we need to add  $32 \times 32 \times 3$  weights to the network. Assuming that we want a layer with the same width and height as the input, we would need to train  $32 \times 32 \times 3$  times  $32 \times 32 = 3,145,728$  weights for just the first layer [15],[16], [17]. This amount is too high for the efficient training of the network. The distinctive feature of CNN is that at least one of the hidden layers is a convolutional layer. A convolutional layer is especially good in detecting patterns and reducing the amount of training parameters.

Firstly, filter  $f$  is initialized at random with a predetermined size and can be considered as matrix of the dimension  $n \times n$ . Secondly, the dot-product of the first  $n \times n$  pixels from the image and the filter is calculated. Finally, this outcome will be the first entry of the output matrix. Like a sliding window, the next  $n \times n$  pixels are selected for the dot-product calculation. An example of a convolutional layer is shown in Figure 2.

In contrast to a classical MLP, for a CNN the weights are fixed locally for the entire next layer, so that the only weights that need to be trained are those of the filter. Assuming we have 6 filters of size  $3 \times 3 \times 3$ , we need only train  $6 \times 3 \times 3 \times 3 = 2,786$  weights instead of approximately 3 millions required to train the MLP.

Since the feature filter is applied locally, the model has the ability to detect and recognize features regardless of the position in the image. This is the first step we see in Figure 1. In

this step an input image of 32x32 is transformed to a layer of 6 filters of 28x28. The first convolutional layer usually focus on the extraction of simple features, such as horizontal lines or simple shapes, whereas deeper layers are capable of extracting more complex features.

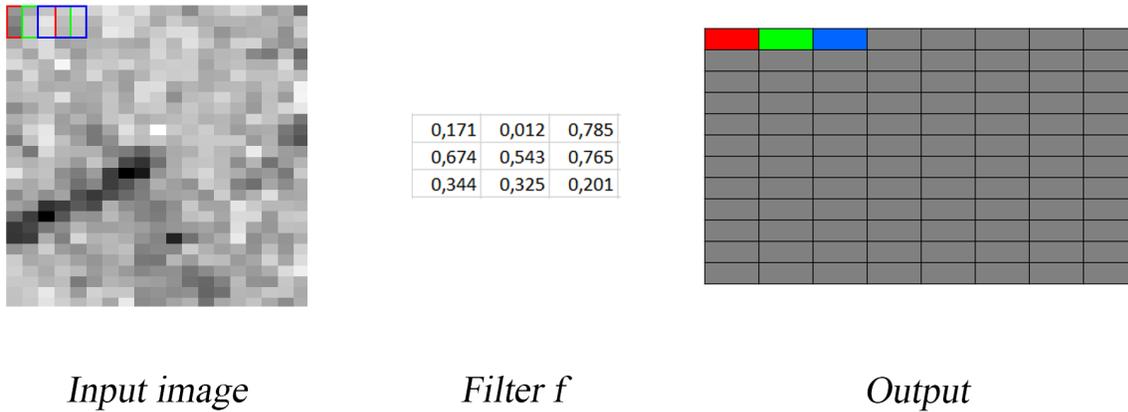


Figure 2. An example of a convolutional layer with a single filter. On the left the input image is presented. Each pixel is in a gray-scale and holds a value between  $[0, 1]$ . Then first 3x3 pixels are selected and multiplied by the filter which then results in the first entry for the output matrix (red). The second and third steps are represented by green and blue colors respectively. This process will continue like a sliding window until the entire output matrix is generated.

### 2.1.2 Pooling

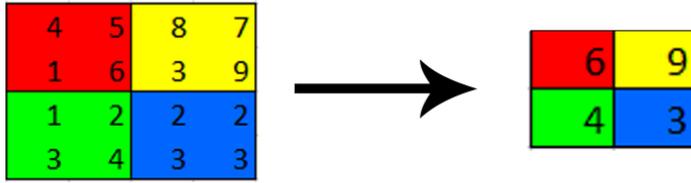
The next step of the CNN is to reduce the size of the filter using Pooling. Pooling down is a down-sampling aiming at the reduction of the dimensional complexity for the next layer. In Figure 3a we can see an example of max-pooling. The highest value from the given area is selected and is used to produce a new matrix accordingly. Another approach of pooling is average-pooling where the average value is selected instead of the maximal value which is illustrated in Figure 3b. Pooling can be compared to the resolution reduction in the area of image analysis. In Figure 1 the second step illustrates the procedure of max-pooling. The reduction of the data dimension from 6x28x28 filters to 6x14x14 filters is being made at this step.

The procedure of sequential application of convolutional layers and pooling is repeated until more distinct feature maps are produced.

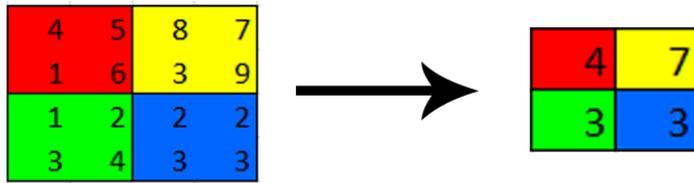
### 2.1.3 Final layer and Activation functions

As a final step fully connected layers are added for classification, similar to that of a MLP. The fully connected layers for classification will be the most time consuming of all steps of CNN training [17]. The amount of output nodes corresponds to the amount of classes we want to identify. An activation function is used to normalize the output and determine the class.

There exist a wide variety of activation functions[18]. Here we introduce only those that are used by the presented models. For multiple output nodes Softmax can be used. The Softmax



(a) An example of max-pooling



(b) An example of average-pooling

Figure 3. Two examples of pooling: max-pooling(upper figure) and average-pooling(bottom figure). Pooling reduces the size of the output layer to reduce complexity.

function takes as input a vector  $z$  of  $K$  real numbers, and normalizes it into a probability distribution consisting of  $K$  probabilities proportional to the exponentials of the input numbers. After the Softmax transformation the sum of all values of the output nodes is 1 and the highest value of these output nodes is selected as the classification node.

For binary classification the sigmoid activation function can be used. Here the CNN only has 1 output node, and the sigmoid activation function will return a value between 0 and 1. The sigmoid function can take any value as input, has the following characteristics:

$$sigmoid(x) = \begin{cases} <0.5 & \text{if } x < 0 \\ >0.5 & \text{if } x > 0 \\ 0.5 & \text{if } x = 0 \end{cases}$$

Therefore, we can determine the class by comparing whether the value of the output node is either bigger or smaller than 0.5. The plot of a sigmoid function is presented in Figure 4.

#### 2.1.4 Overfitting

A common problem with artificial neural networks is overfitting. The overfitting occurs when a model is trained for a long time on the same data. The model will be very good at identifying the classes of the training data, but on unseen data it will perform poorly. Instead of learning distinct features belonging to a class, the model will learn to recognize the already seen during the training procedure images and classify accordingly. Therefore, not being able to correctly

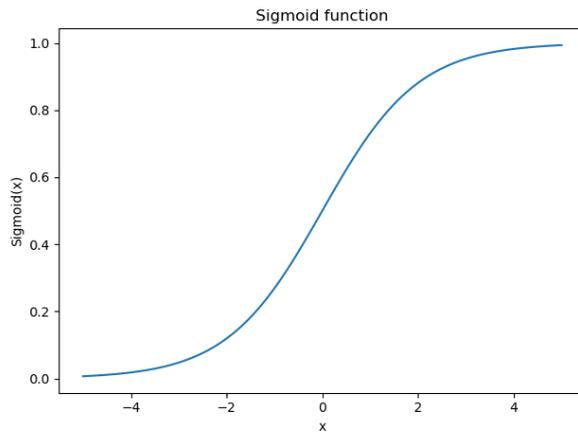


Figure 4. A plot of the sigmoid function.

classify unseen images. Having a very high accuracy while training and a low accuracy when testing implies overfitting. Since having a bigger amount of training examples can improve the generalization properties of the network, overfitting can be avoided.

### 2.1.5 Network training

During training the network "learns" by optimizing its weights and biases in the network through the process of backpropagation. When all training examples of a batch have been passed through the network, we compare the received output with the true labels and use gradient descent to modify the weights of the model so the difference is minimized. We can define the training procedure as follows: Consider a training set  $D_N$  composed of  $N$  training examples,

$$D_N = \{x_i, y_i\} 1 \leq i \leq N \quad \forall (x_i, y_i) \in D_N :$$

- compute  $\Delta\theta = \nabla_{\theta} J(\theta)$
- Update:  $\theta = \theta - \mu\Delta\theta$

Where  $\mu$  is the learning rate,  $J(\theta)$  is the objective function parameterized by model's parameters  $\theta$ .

The learning rate is a hyper parameter that can be optimized. The gradients in a neural network may differ greatly, which will make it difficult to have a single global learning rate. A solution for this is having a signed weight for each layer. This means that for every layer we have a different learning rate. Another option to improve the learning rate is to adjust the learning rate throughout the training [19]. The adaptive learning rate take into consideration the gradients from previous step to determine whether the learning rate should increase or if the learning rate should decrease. RMSprop combines both the adaptive learning rate and the signed learning rate[20].

## 2.2 Models

For this paper we introduce two models that we will use for the experiments. The first model is the LeNet model[14]. This model has proven to be very powerful in identifying handwritten

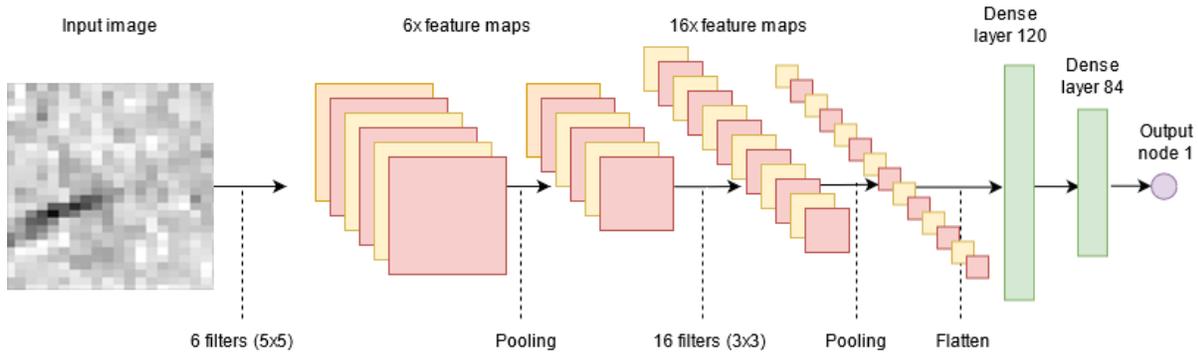


Figure 5. The design of LeNet as used for our experiments

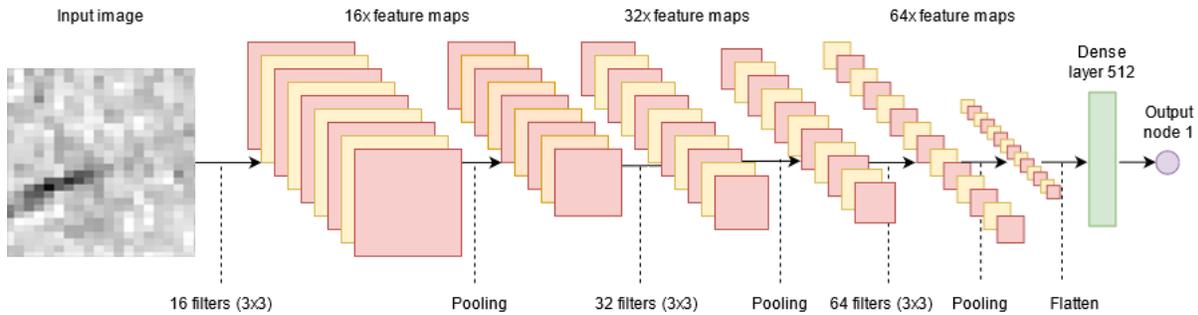


Figure 6. Phan-model as used for our experiments

numbers. While this problem is different, the input data is of a similar format. Therefore, using the architecture of LeNet is a good baseline for testing. Due to the fact that the classification task is different from the original task, the output layer of the model used in this work is not the same as was used in the original version of the architecture. In the original model handwritten numbers were identified. As there are ten different digits the output layer needs to be adjusted for binary classification. Instead of having ten output nodes and a softmax activation function, a single output node with sigmoid activation function is used.

The original LeNet used max-pooling for the reduction of complexity, however, average-pooling could also be used instead. In this paper we tested both variants. A full design can be seen in Figure 5.

The next model is the Phan-model from [21] where in the original version the goal of the CNN was a binary classification between pictures of grass and pictures of dandelions. Here we needed to adjust the input layer to accept greyscale images instead of colored images. The model has a bigger amount of feature maps, and convolutional layers than the LeNet model. This model also uses a single output node with sigmoid activation. A full setup of this model can be seen in Figure 6. Similarly to the LeNet model we have two versions, the first uses max-pooling like in the original experiments and the second version uses average-pooling. All models were optimized using RMSprop.

Because this classification problem is relatively simple, we have chosen to use architectures that are not as complex. Having architectures with a lower complexity will reduce the amount of training time and data required, while still giving a good result.

## 2.3 Required data

### 2.3.1 TROPOMI/Sentinel-5

The most important part of the data for our experiments comes from the Tropospheric Monitoring Instrument (TROPOMI), onboard the Sentinel-5 Precursor (S5P) satellite[22]. It provides measurements of atmospheric trace gases as well as cloud and aerosol properties at an unprecedented spatial resolution of approximately  $5.5 \times 3.5 \cdot \text{km}^2$ , achieving global coverage in 1 day[23]. The data product used in this paper is  $\text{NO}_2$  tropospheric column. We focus on the eastern Mediterranean Sea. The area was chosen for several reasons. First, it is close to the equator thus we do not need to worry about regridding and distorting the map too much. For more details about the regridding see Chapter 2.4. Secondly, it covers a part of one of the busiest trading routes. This route goes through the Suez Canal connecting Europe and Asia. To demonstrate how busy this route is, when the Suez Canal was obstructed for six days the estimated losses were roughly \$9.6bn(€7.9bn) per day [24].

In previous study, in order to ensure the good distinguishability of ship plumes, a day with excellent weather conditions (cloud-free skies, low winds and sun glint) was chosen[12]. For gathering a substantial amount of data needed for CNN training we do not have the luxury of choosing the best days. Therefore, the overall quality of the used images will be lower than what was presented in a previous study. We did, however, choose to pick dates after March 2019, since after that date ECMWF wind data (horizontal and vertical wind speeds) is available as a support product in the TROPOMI data file. To ensure a wide variety of analyzed scenes, seven months of data, covering the period from April to October 2019 were used for the analysis. The wind data is necessary for our study as by associating a detected plume with the track of the ship, it allows to separate ship plumes from the plumes of different origin.

### 2.3.2 AIS

Most ships are required to carry on board an Automated Identification System(AIS). This allows official institutions to monitor the position of each ship. The AIS data is not available publicly. For this study the data was provided by the Inspectorate for Human Environment and Transport (ILT) of the Netherlands. Due to the privacy reasons, the provided data does not contain the ship identifiers, so all analyzed ships were anonymous.

We need the shipping data to determine if a visible plume is produced by an aquatic vessel. Our objective is to find out whether the  $\text{NO}_2$  plume produced by an analysed ship can be identified visually. Gaseous clouds disperse over time due to the wind. Therefore, we only use shipping data that is at most 2 hours before the overpass. With an overpass we mean the time the TROPOMI/S5P passed over the given area. With that we can plot the location and route where the ships were right before an overpass.

The location of a plume does not exactly corresponds with the location of the ship-emitter. It is heavily influenced by the wind speed and direction. As it was mentioned in the previous section, the TROPOMI data also holds the metadata for the horizontal and vertical wind speeds. To determine where the plume is expected to be, a simple transformation is applied.  $s$  is the distance between the original position and the expected position of the plume in the direction of the wind in km which is calculated the following formula:  $s = u \cdot |\Delta t|$ , with  $u$  the wind speed in km/h, and  $|\Delta t|$  the difference between the AIS time and the time of the Sentinel-5 overpass in hours. We want to exclude aquatic vessels that will most likely not produce a visual plume, such as leisure ships or fishing boats to reduce the amount of ships

that need to be analyzed. In an attempt to balance the ratio of plume producing ships, we remove all ships with a speed lower than 15 kn(knots).

## 2.4 Regridding

The data from the TROPOMI/S5P is geographical data and is projected on a sphere. In order to apply all the image processing techniques we need to project the spherical data to a 2-D map. This process is called regridding and is widely used in geographical data. The chosen area is between 15-31 longitude and 30-36 latitude. Since we know that the TROPOMI/S5P has a spacial resolution of  $5.5 \times 3.5 \cdot \text{km}^2$  and the area is 667km by 1538km we projected to entire area by applying the size of an individual grid equal to 174 by 304 pixels of size  $5.06 \times 3.83 \cdot \text{km}^2$ .

## 2.5 Pre-processing

Pre-processing or data cleaning is a crucial step in machine learning. Removing noisy data, handling missing data and removing outliers are all part of data pre-processing. In this paper we are interested in the recognition of NO<sub>x</sub> plumes coming from ships. Nitrogen levels are far denser above city areas than above the open sea. In spite of the fact that our chosen area is largely above the Mediterranean Sea, some port areas are still included in our selected data. In these port areas the nitrogen levels measured by the TROPOMI/S5P are much higher than levels in the plumes we are interested in.

In the paper [12] the authors use values in the range of [0.5, 2.5] to assure visual distinguishability of ship plumes. We can assume that any value much higher than 2.5 is mostly produced by land-based sources, and not by an aquatic vessel. In addition to having too high valued data points we are also not interested in the variability of low values. Having a range that is too large will result in images where plumes are not visible enough for humans to label. This is because the differences between background values and ship produced plumes are relatively low, when compared to the differences of background values and land-based plumes. To manage this particular issue we use clipping.

Clipping is the term used for transforming data between an minimum and a maximum. If a data point is higher than a predetermined maximum, this data point is clipped/lowered to the maximum value, in our case 2.5. The same principle is applied for the minimum value. Here the value is increased to the minimum value. To ensure the comparability of analysed images, it is important to make sure that the data is restricted to the same range. With clipping we lose a significant part of the information, since apply a transformation that cannot be reversed. However, since we are only interested in finding high concentration plumes, we do not need to differentiate between high and super-high concentration plumes. Similarly we are not interested in plumes with concentrations below the average background value, that is usually between 0.5 and 0.7 depending on the day. Therefore, clipping is still a desired transformation.

The next step in pre-processing is to apply a filter, by applying a filter we attempt to improve the features of the image data by suppressing unwanted distortions. Furthermore, we aim to enhance some important image features so that it will benefit the CNN model. We have 5 different filters for data enhancement.

### 2.5.1 Mean

The first filter we apply is the mean filter. This is probably the simplest filter to reduce noise. Like a sliding window we calculate the center by taking the mean of all values in the

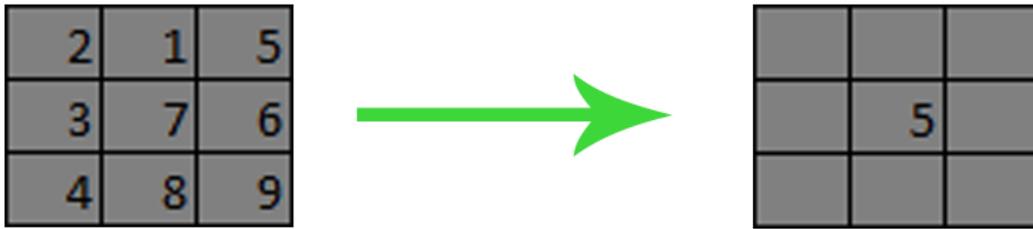


Figure 7. Application of a mean filter on an image. The center square is calculated by taking the mean of all the values in the kernel. Like a sliding window, this process is repeated until the area of the entire image is covered.

window. The shape of the used kernel is usually a square but can be of any shape or form. In the example presented in Figure 7 a 3x3 kernel is used.

### 2.5.2 Median

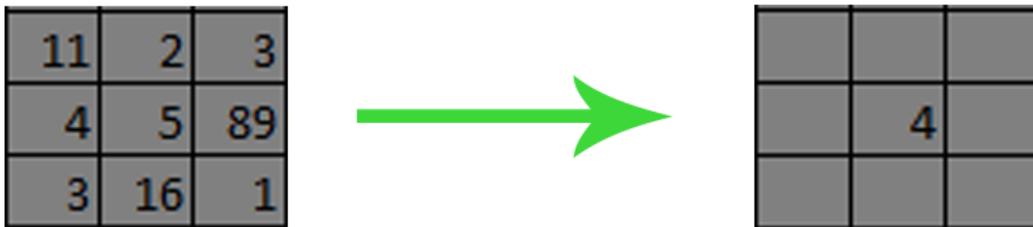


Figure 8. Application a median filter on an image. The center square is calculated by taking the median of all the values in the kernel. Like a sliding window, this process is repeated until the area of the entire image is covered.

The median filter is very similar to the mean filter, except it works better on data containing outliers. Like the mean filter a kernel is used to calculate the center value. But this time the median is selected instead of the mean. This will prevent a single outlier to heavily influence the value of the new image. In Figure 8 we see an example of how the median is applied.

### 2.5.3 Gaussian

Similarly to both the above-discussed filters, the Gaussian filter also uses a kernel and a sliding window. However, instead of taking the mean or median we use a filter with a Gaussian

$$\frac{1}{273} \cdot \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline \end{array}$$

Figure 9. An example of a Gaussian filter with a  $\sigma = 1.0$

distribution. The probability density function of the 1-D Gaussian where  $\sigma$  is one standard deviation and the distribution's mean is centered around 0 is defined as follows:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

We need a distribution around 0 to make sure that the values after the Gaussian filter do not differ too much from the original values. For a 2-D Gaussian distribution the form is:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

If we want a perfect Gaussian distribution filter, we need an infinite kernel-size which will have non-zero values everywhere. However, in practice if we choose to have a kernel with three standard deviations from the center, a 5x5 kernel, we can consider everything outside of that kernel as zero, since these kernel values will be very small anyway. In Figure 9 we see an example of a Gaussian filter with a standard deviation( $\sigma$ ) of 1.0.

#### 2.5.4 Bilateral

Gaussian filtering will blur the image uniformly, no matter what the value of the neighboring pixels are. Bilateral filtering is similar to Gaussian filtering with the addition of a modulating function that compares the values of the central pixel and its neighbouring pixels. The modulating function will produce a value in range  $(0, 1]$  depending on the similarity between the pixels' values.

- If the central pixel and the neighbouring pixel are of a similar value, the modulating function will be close to 1 and thus lose to the value obtained from the Gaussian filter.
- If the pixel values are very different, the modulating function will be close to 0, effectively turning off Gaussian blur for this pixel.

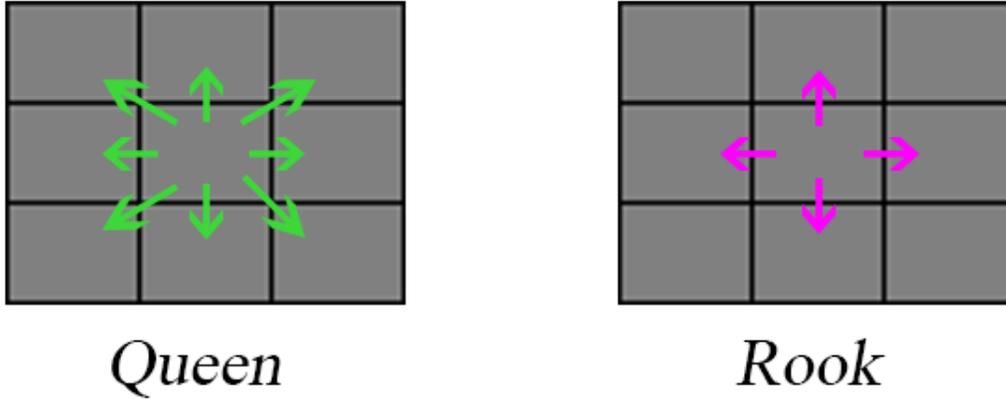


Figure 10. Types of contiguity with the Queen on the left and the Rook on the right. In the Queen contiguity matrix all eight pixels are considered its neighbours. While the Rook only has four neighbours.

### 2.5.5 Moran's I

Moran's I is not a typical filter as the previously discussed filters are, but is a spacial auto-correlation technique based on both feature correlation and feature values simultaneously. It was first proposed by [25]. Given a set of features and an associated attribute, it evaluates whether the pattern expressed is clustered, dispersed, or random. In this paper we are interested in finding clusters of high values.

The local Moran's I for a pixel  $i$  is calculated by the following formula:

$$I_i = \frac{x_i - \bar{x}}{S_i^2} \sum_{j=1, j \neq i}^n w_{ij}(x_j - \bar{x})$$

Where

- $x_i$  is the respective pixel value
- $\bar{x}$  is the mean of all pixels in the analyzed image
- $S_i^2 = \frac{\sum_i (x_i - \bar{x})^2}{n}$  is an estimation of the variance of all pixels in the image
- $n$  is the number of pixels
- $w_{ij}$  the contiguity matrix

There are two type of contiguity matrices, the Rook and the Queen named after their movement as chess pieces as shown in Figure 10. With the Queen contiguity matrix all of the surrounding pixels are considered neighbouring pixels, while the Rook matrix only has the horizontal and vertical neighbours.

To illustrate the variation between the different pre-processed images examples are shown in Figure 11.

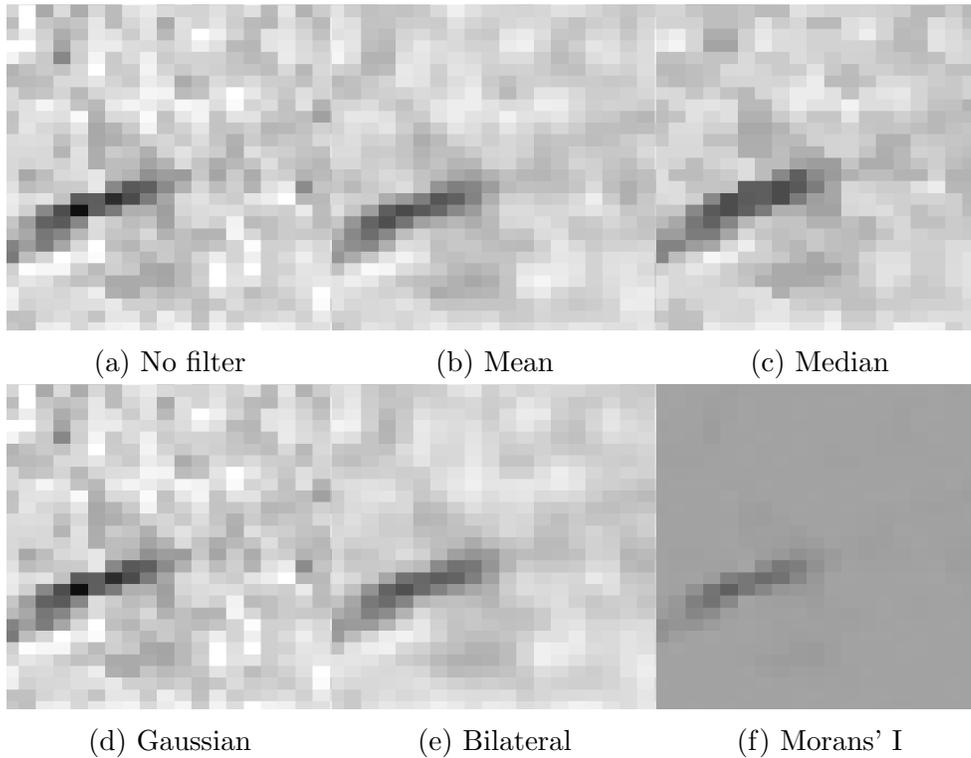


Figure 11. Examples of the different pre-processing techniques on the same image data

### 3 Data

To train the CNN a the labeled data set is required. The labeling must be done by hand and in this section we will explain the selection process. To produce images of one day for the dataset the following steps should be undertaken:

1. Select the TROPOMI/5SP data for a given date
2. Regrid the data to a 2-D plane
3. Clip the data to prevent dispersed data
4. Replace missing (nan) values with 0
5. Apply one of the pre-processing techniques
6. Remove slow ships (speed  $\leq 15$  kn)
7. Use the AIS-data and the wind speed data to project where a ship might produce a plume
8. Produce an image for the entire area
9. For each ship
  - Plot the areas around a single ship with size 1.0x1.0(latitude x longitude)
  - Check if there is more than 90% overlap between areas. If this is the case we remove it from dataset, to prevent (almost) duplicate images

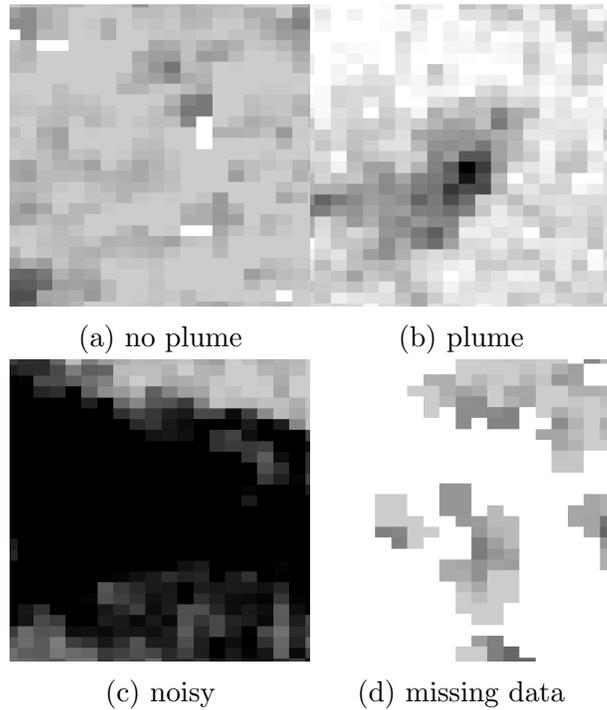


Figure 12. Four examples of images we generated. In the first image there is no visible plume, and only a small portion of the data missing. Therefore, it is labelled " no plume". In the second image a plume is clearly visible, so we label it "plume". The third image is too close to a land-based source and therefore it can not be labelled. The last image has large parts of missing data, so we have no label.

### 3.1 Data selection

After this process we are left with images containing possibly visible plumes that need to be labelled. The labelling is done by visual analysis of the three following images. First, the raw image that we will need to feed to the network. The second image is the same image, but with the ships projected on the image, since we need that to determine whether a plume is produced by a ship. The final shown image is the entire selected area. The latter is needed in order to obtain a wider perspective of the position of the ship and observe that the analyzed plume is not of land origin. An example of the images used for the labelling procedure is shown in Figure 13.

There are 3 possible options when shown an image. The first is to add this to the list of visible plumes. The second option is to label the images as not containing a visible plume. The third option is to discard the image, since it has a lot of missing data, or it is too noisy. Most images end up in the final category. In Figure 12 we see the four possible options. We have an image where there is no clear plume, an image where there is a clear plume, an example of an image that has too much noise and an image where there is too much data missing. In some cases, only a small part of the data is missing, then we can decide to label the image anyway and replace the missing data with white pixels (value of 0).

During the process of selecting the data it is possible to rotate between all of the differently pre-processed images to determine in what category the image should be placed.

Out of the 5.590 generated images, 649 images of visible plumes and 708 images of no visible plume were selected.

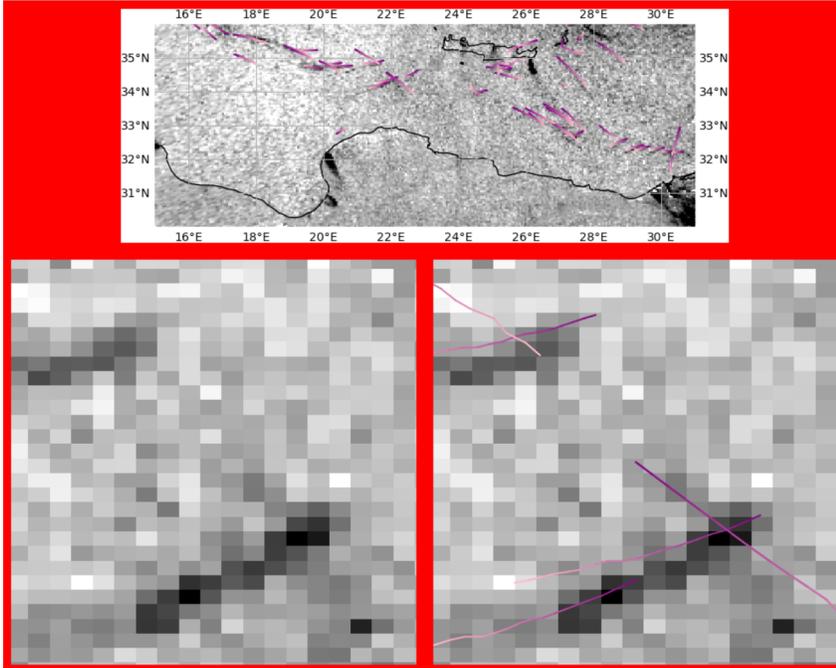


Figure 13. The setup developed for manual data labelling. Each generated image must be labelled "plume", "no\_plume" or discarded. The  $\text{NO}_2$  values are represented by the gray scale image. The wind shifted shipping tracks are plotted from purple to white, with purple being the most recent location.

	training set	validation set	test set
no plume	417	140	151
plume	397	131	121
total	814	271	272

TABLE I. The spread of the training(60%)/validation(20%)/test(20%) set

## 4 Classifier

In order to train the classifier the labelled data must be split into three distinct sets. The training set, the validation set and the test set. Firstly, the model will be trained on the training set. The validation set is used for the optimization of the hyper parameters of a model during the training. Finally, the test set is used for testing on completely unseen data. Using only a validation set as unseen data will give a biased result as the hyper parameters are optimized for the validation set. The training/validation/test sets are split into 60%, 20% and 20% respectively. All the data is from the period between 01-04-2019 and 31-10-2019. The data distribution of the images per month is shown in Figure 14. It is clear that during the summer months the amount of good examples is higher than in the other months. This is due to the fact that these months generally have clearer skies, resulting in higher quality images. Therefore, the amount of good examples that can be found during these months are greater. For an example of a full test set see the Appendix A.

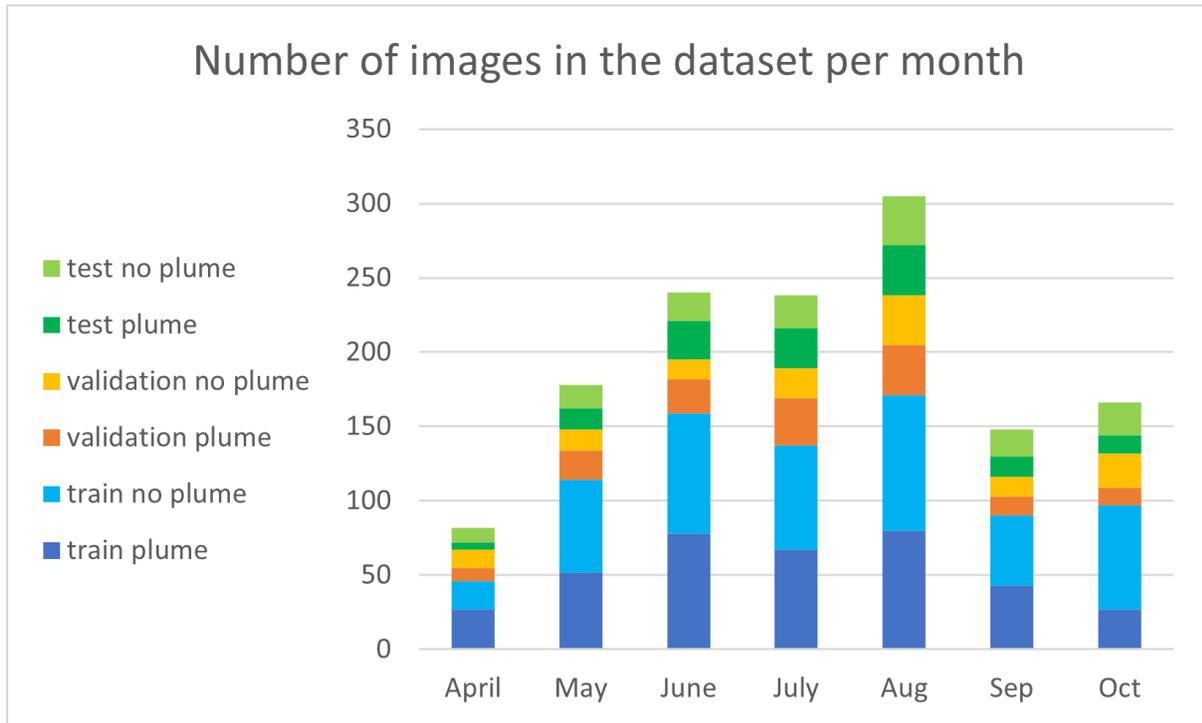


Figure 14. The amount of images in the dataset per month

	no filter	mean	median	Gaussian	bilateral	Moran's' I	average
LeNet(max)	90.10	91.77	89.38	<b>92.82</b>	92.17	92.17	91.06
LeNet(avg)	92.70	<b>94.58</b>	93.58	92.70	94.34	91.43	93.22
Phan(max)	93.92	94.68	<b>94.77</b>	93.92	93.90	91.74	93,82
Phan(avg)	94.56	95.08	93.18	94.56	<b>95.99</b>	94.36	94,62
average	92.82	94.03	92.73	92.82	94.26	92.43	

TABLE II. The results from training four models for each filter type. Each model was trained 10 times and the average accuracy of correctly classified images is shown in (%).

## 4.1 Results

We trained each of the models with a different pooling technique and applying different methods for data filtering for 50 epochs with 15 steps per epoch. To ensure comparability between the results, we have fixed the initialization of each model for every data filtering technique. All models are trained on the same distribution of the training/validation/test set once, and as always with the same initialization. Afterwards the dataset is reshuffled into a new training, validation and test sets. The process of reshuffling the data and training of each model/filtering combination is repeated 10 times and the average results are shown in Table II. Because the initial dataset is well balanced it is almost certain that each part of the dataset will also be close to balanced.

Since we have removed all ships with a speed lower than 15 kn in an attempt to balance the ratio of plume producing ships, it would be interesting to compare the speeds of ships labelled with "plume" and ships labelled with "no plume". In Figure 15 we see the histogram for the speeds of each label. The average speed for "plume" is 17.78 and the average speed for "no plume" is 17.08.

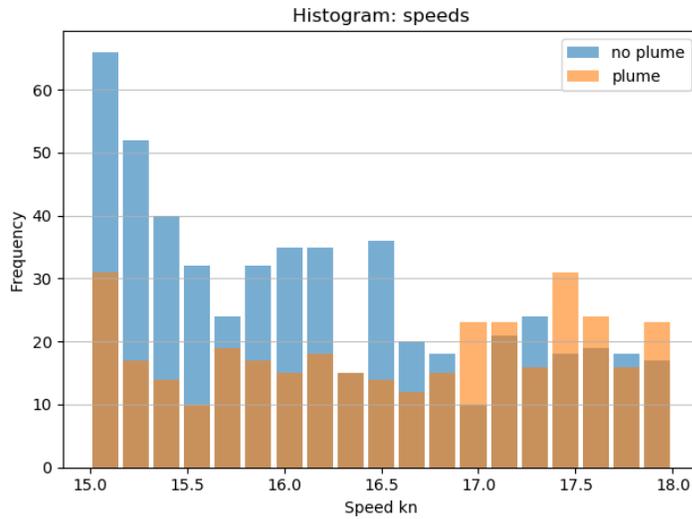


Figure 15. Histogram of the speed of the ships per label

## 5 Discussion

In Table II we can see all the results from the trained models. The first thing one can notice is that the highest accuracy was reached by the Phan-model using average-pooling and the bilateral filter applied. With a score of 95.99% it slightly outperforms all other combinations. The second highest score was also the Phan-model with average-pooling and a mean filter. Even though all results are relatively close, the Phan-model slightly outperforms the LeNet-model, likely due to the fact that it has a higher amount of feature maps.

We can see that over average mean and bilateral filter are the best scoring filters. All of the other filters have a similar result. Moran's I has great potential since it is really good in smoothing out the randomness of the data and outline the clusters in a way that a convolutional layer could not do. However, it also outlines the low-value clusters which we are not interested in. This could cause confusion for the CNN model in classification especially in producing false positives.

All results are relatively high and close together, this could indicate that the classification task was a relatively simple one.

Another important note is that, we were not able to add all of the generated images to the data set. A substantial amount of images are polluted by  $\text{NO}_2$  produced on land. A lot of data is also missing due to clouds that block the satellite or other measuring obstructions. All of this results in a fact that a substantial amount generated data could not be labelled and thus will not be added to the data set. Out of the generated 5500 images, roughly 1400 were labelled.

In addition, it must be noted that during the selection process we aimed for a balanced set. Because of this some images where there were no visible  $\text{NO}_2$  plumes were also not labelled. Preferably even more data should be produced and labelled to produce a larger and more robust dataset.

Figure 14 shows the distribution of the dataset over each month. It is clear that during the summer months the amount of good examples is higher than in the other months. This is due to the fact that clear skies are more common during the summer. Having clear skies ensures good quality of images. Therefore, it is more likely that good examples can be found.

We removed all ships that had a speed lower than 15 kn from the dataset. We did this because it would remove a lot of ships that were most likely not producing a visible plume and it helps to achieve a balanced dataset. In Figure 15 we see the distribution of the speed of the ships that we labelled. A greater speed does indicate a higher chance of producing a plume, but due to the fact that there is a lot of overlap between the two classes, it is not definitive. The average speed of the ships labelled "plume" is a bit higher: 17.78 for "plume" and 17.08 for "no plume". It is interesting that the distribution of the classes are really different however. While the speed for "plume" seems to be distributed relatively uniformly, the speed for "no plume" seems to cluster more around lower speeds, with the highest cluster in the lowest speed range. This could indicate that if we included speeds lower than 15 kn, a lot more "no plume" ships could be discovered.

In this paper we used the eastern Mediterranean Sea mainly because of its high quality imaging. Other areas do not have the same features and should be researched with slight alterations. The Mediterranean Sea generally has good weather conditions for satellite measurements, other areas might have more missing data due to clouded skies or other factors that influence the satellite measurements.

## 6 Conclusion

In this paper we explored the possibility to automatically detect plumes generated by ships in the Mediterranean Sea. We gathered the required data and generated images possibly containing NO<sub>2</sub> plumes and applied multiple image pre-processing techniques to the data. After creating these images, we developed a setup for an efficient data labeling and then we labelled the data with either "plume" or "no plume". Afterwards, we train the CNN models on the dataset and produce the results. It turns out that the bilateral filter gives the best results, with the mean filter as a close second. We see that applying no filter will produce similar results as all other filters. Moran's I filter performs the worst, however, all results are very close to each other. The reason all results are so close could be due to the fact that the classification task is a relatively easy one. The best combination was the Phan model with average pooling and a bilateral filter. Resulting in an accuracy of 95.99%. These results are fairly good. However, we must note that the images were carefully selected and labeled. Images that had too much missing data, or were too noisy were removed from the dataset. Removing these images of low quality resulted in a very clean dataset that was good for training a CNN model.

Overall this study has indicated that the problem of automatically detecting NO<sub>2</sub> plumes produced by ships and measured by satellite sensor can be solved by a CNN. However, more research is needed before this method can be used for the detection of high NO<sub>2</sub> producing ships on a global scale.

## 7 Future research

While the main focus of this paper was exploring whether it is possible to produce a model that can classify between a visible plume and a plume not distinguishable from the background, the bulk of this paper was to produce and label the data set the dataset. Therefore, the amount of different models trained is low. Future research could be exploring different model designs that include more hyper parameter optimization. It would also be interesting to apply different

auto-correlation techniques on the data before producing an image. Moran's I worked relatively well seeing how much different the Moran's I-filtered images were. It could be interesting to see if a model can be created that could distinguish between a clear image, and an unclear one. Then that model could be used to filter out such unwanted images, before applying this model.

## References

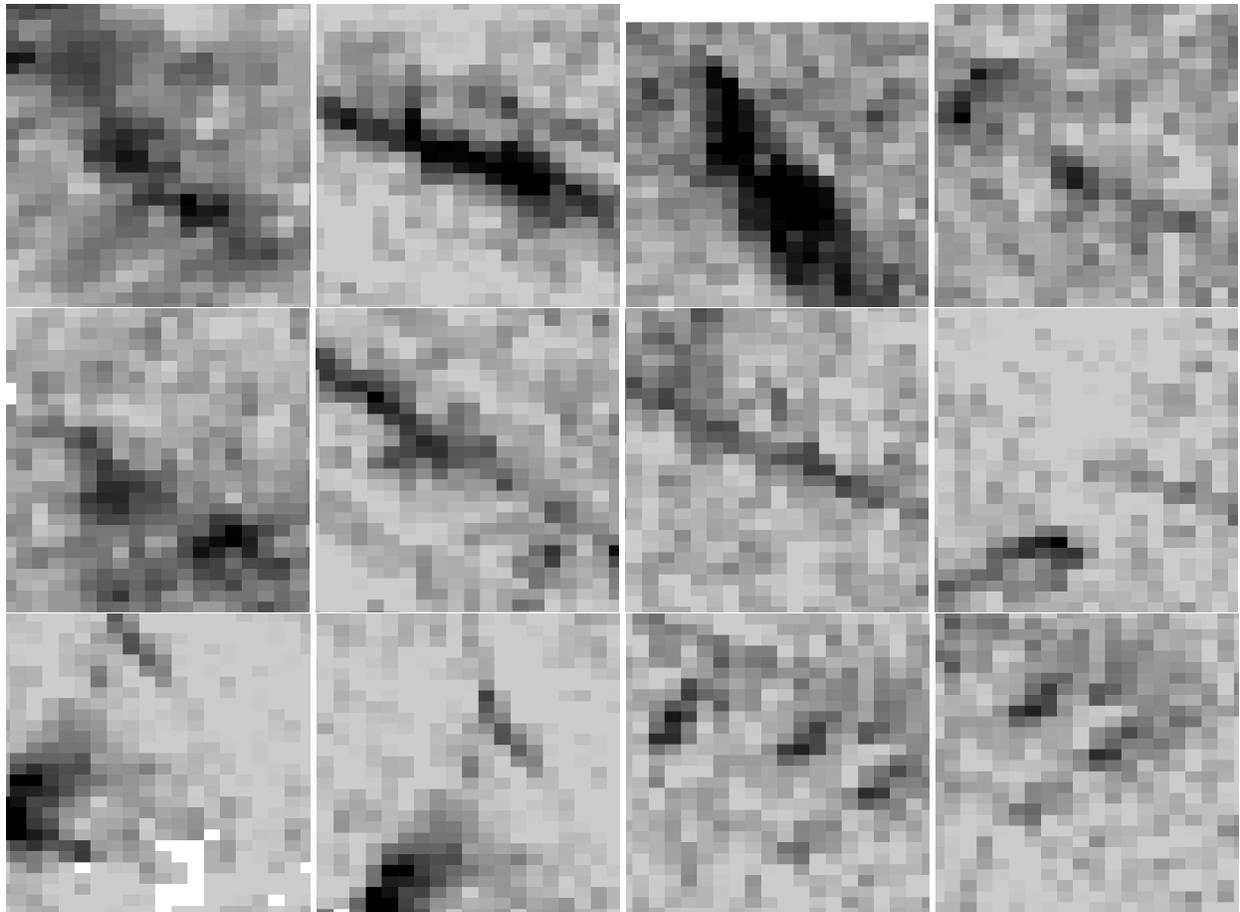
- [1] International Maritime Organization. *Nitrogen Oxides (NO<sub>x</sub>) – Regulation 13*. URL: [https://www.imo.org/en/OurWork/Environment/Pages/Nitrogen-oxides-\(NOx\)-%5CE2%5C%80%5C%93-Regulation-13.aspx](https://www.imo.org/en/OurWork/Environment/Pages/Nitrogen-oxides-(NOx)-%5CE2%5C%80%5C%93-Regulation-13.aspx).
- [2] International Maritime Organization. *Sulphur oxides (SO<sub>x</sub>) and Particulate Matter (PM) – Regulation 14*. URL: [https://www.imo.org/en/OurWork/Environment/Pages/Sulphur-oxides-\(SOx\)-%5CE2%5C%80%5C%93-Regulation-14.aspx](https://www.imo.org/en/OurWork/Environment/Pages/Sulphur-oxides-(SOx)-%5CE2%5C%80%5C%93-Regulation-14.aspx).
- [3] Kevin Cullinane and Rickard Bergqvist. “Emission control areas and their impact on maritime transport”. In: *Transportation Research Part D: Transport and Environment* 28 (2014). Emission Control Areas and their Impact on Maritime Transport, pp. 1–5. ISSN: 1361-9209. DOI: <https://doi.org/10.1016/j.trd.2013.12.004>. URL: <https://www.sciencedirect.com/science/article/pii/S136192091300148X>.
- [4] Harshit Agrawal et al. “In-use gaseous and particulate matter emissions from a modern ocean going container vessel”. In: *Atmospheric Environment* 42.21 (2008), pp. 5504–5510. ISSN: 1352-2310. DOI: <https://doi.org/10.1016/j.atmosenv.2008.02.053>. URL: <https://www.sciencedirect.com/science/article/pii/S1352231008002057>.
- [5] W. V. Roy and Kobe Scheldeman. “Best Practices Airborne MARPOL Annex VI Monitoring”. In: 2016.
- [6] J. Beecken et al. “Airborne emission measurements of SO<sub>2</sub>, NO<sub>x</sub> and particles from individual ships using a sniffer technique”. In: *Atmospheric Measurement Techniques* 7.7 (2014), pp. 1957–1968. DOI: 10.5194/amt-7-1957-2014. URL: <https://amt.copernicus.org/articles/7/1957/2014/>.
- [7] N. Berg et al. “Ship emissions of SO<sub>2</sub> and NO<sub>2</sub>: DOAS measurements from airborne platforms”. In: *Atmospheric Measurement Techniques* 5.5 (2012), pp. 1085–1098. DOI: 10.5194/amt-5-1085-2012. URL: <https://amt.copernicus.org/articles/5/1085/2012/>.
- [8] Daniel A. Lack et al. “Particulate emissions from commercial shipping: Chemical, physical, and optical properties”. In: *Journal of Geophysical Research: Atmospheres* 114.D7 (2009). DOI: <https://doi.org/10.1029/2008JD011300>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2008JD011300>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JD011300>.
- [9] L. Pirjola et al. “Mobile measurements of ship emissions in two harbour areas in Finland”. In: *Atmospheric Measurement Techniques* 7.1 (2014), pp. 149–161. DOI: 10.5194/amt-7-149-2014. URL: <https://amt.copernicus.org/articles/7/149/2014/>.

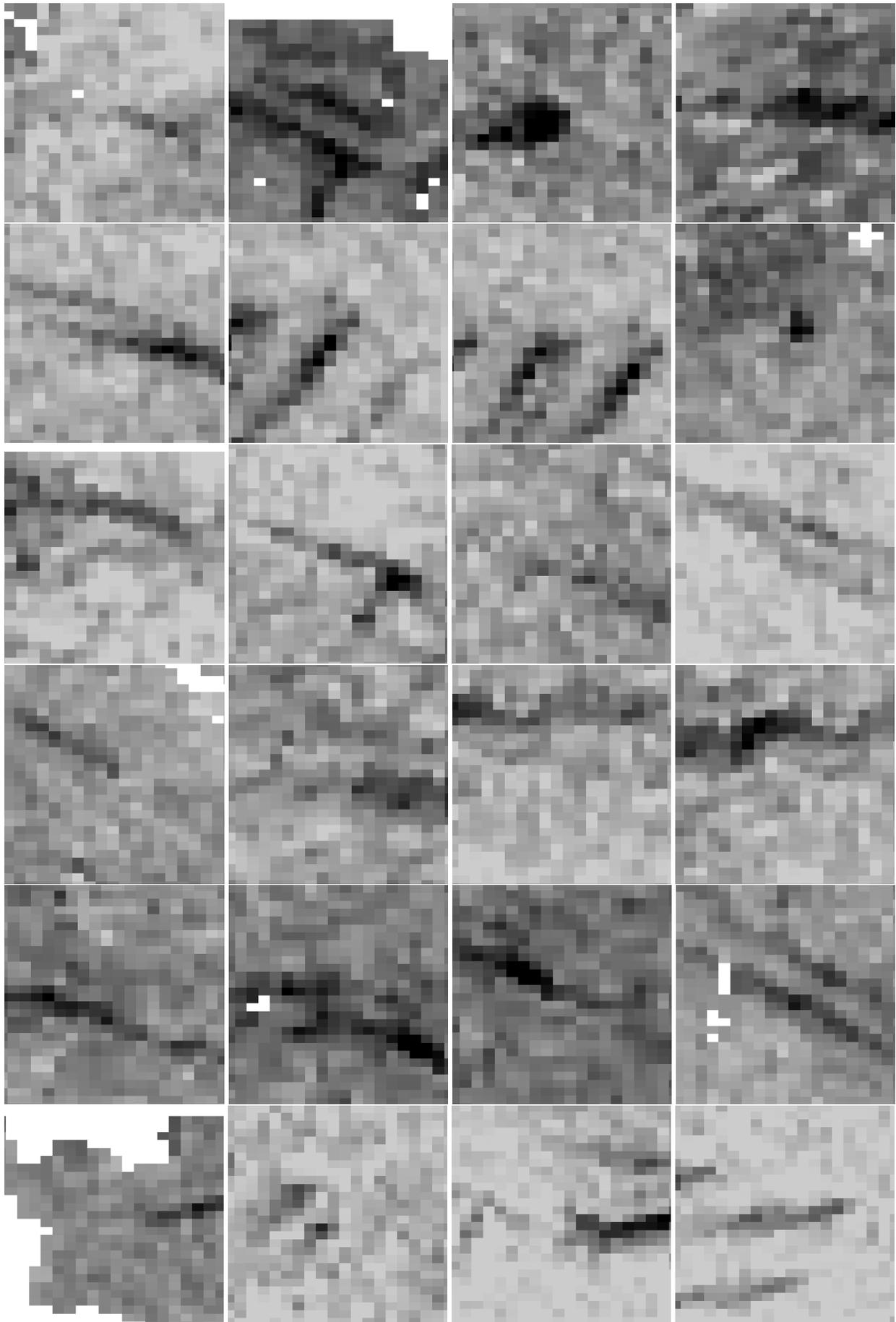
- [10] Robert McLaren et al. “A survey of NO<sub>2</sub>:SO<sub>2</sub> emission ratios measured in marine vessel plumes in the Strait of Georgia”. In: *Atmospheric Environment* 46 (2012), pp. 655–658. ISSN: 1352-2310. DOI: <https://doi.org/10.1016/j.atmosenv.2011.10.044>. URL: <https://www.sciencedirect.com/science/article/pii/S135223101101123X>.
- [11] S.F. Schreier et al. “Ship-based MAX-DOAS measurements of tropospheric NO<sub>2</sub> and SO<sub>2</sub> in the South China and Sulu Sea”. In: *Atmospheric Environment* 102 (2015), pp. 331–343. ISSN: 1352-2310. DOI: <https://doi.org/10.1016/j.atmosenv.2014.12.015>. URL: <https://www.sciencedirect.com/science/article/pii/S1352231014009637>.
- [12] Aristeidis K Georgoulas et al. “Detection of NO<sub>2</sub> pollution plumes from individual ships with the TROPOMI/S5P satellite sensor”. In: *Environmental Research Letters* 15.12 (Dec. 2020), p. 124037. DOI: [10.1088/1748-9326/abc445](https://doi.org/10.1088/1748-9326/abc445). URL: <https://doi.org/10.1088/1748-9326/abc445>.
- [13] Y. LeCun B. Boser J. S. Denker D. Henderson R. E. Howard W. Hubbard L. D. Jackel. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 4 (1989), pp. 541–551. DOI: <https://doi.org/10.1162/neco.1989.1.4.541>.
- [14] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [15] Yanming Guo et al. “Deep learning for visual understanding: A review”. In: *Neurocomputing* 187 (2016), pp. 27–48.
- [16] Jianxin Wu. “Introduction to Convolutional Neural Networks”. In: 2017.
- [17] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. “Understanding of a convolutional neural network”. In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–6. DOI: [10.1109/ICEngTechnol.2017.8308186](https://doi.org/10.1109/ICEngTechnol.2017.8308186).
- [18] Chigozie Nwankpa et al. “Activation Functions: Comparison of trends in Practice and Research for Deep Learning”. In: *CoRR* abs/1811.03378 (2018). arXiv: 1811.03378. URL: <http://arxiv.org/abs/1811.03378>.
- [19] Qing Song, Yilei Wu, and Yeng Chai Soh. “Robust Adaptive Gradient-Descent Training Algorithm for Recurrent Neural Networks in Discrete Time Domain”. In: *IEEE Transactions on Neural Networks* 19.11 (2008), pp. 1841–1853. DOI: [10.1109/TNN.2008.2001923](https://doi.org/10.1109/TNN.2008.2001923).
- [20] Vitaly Bushaev. *Understanding RMSprop - faster neural network learning*. Sept. 2018. URL: <https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a>.
- [21] Binh Phan. *10 Minutes to Building a CNN Binary Image Classifier in TensorFlow*. July 2020. URL: <https://towardsdatascience.com/10-minutes-to-building-a-cnn-binary-image-classifier-in-tensorflow-4e216b2034aa>.

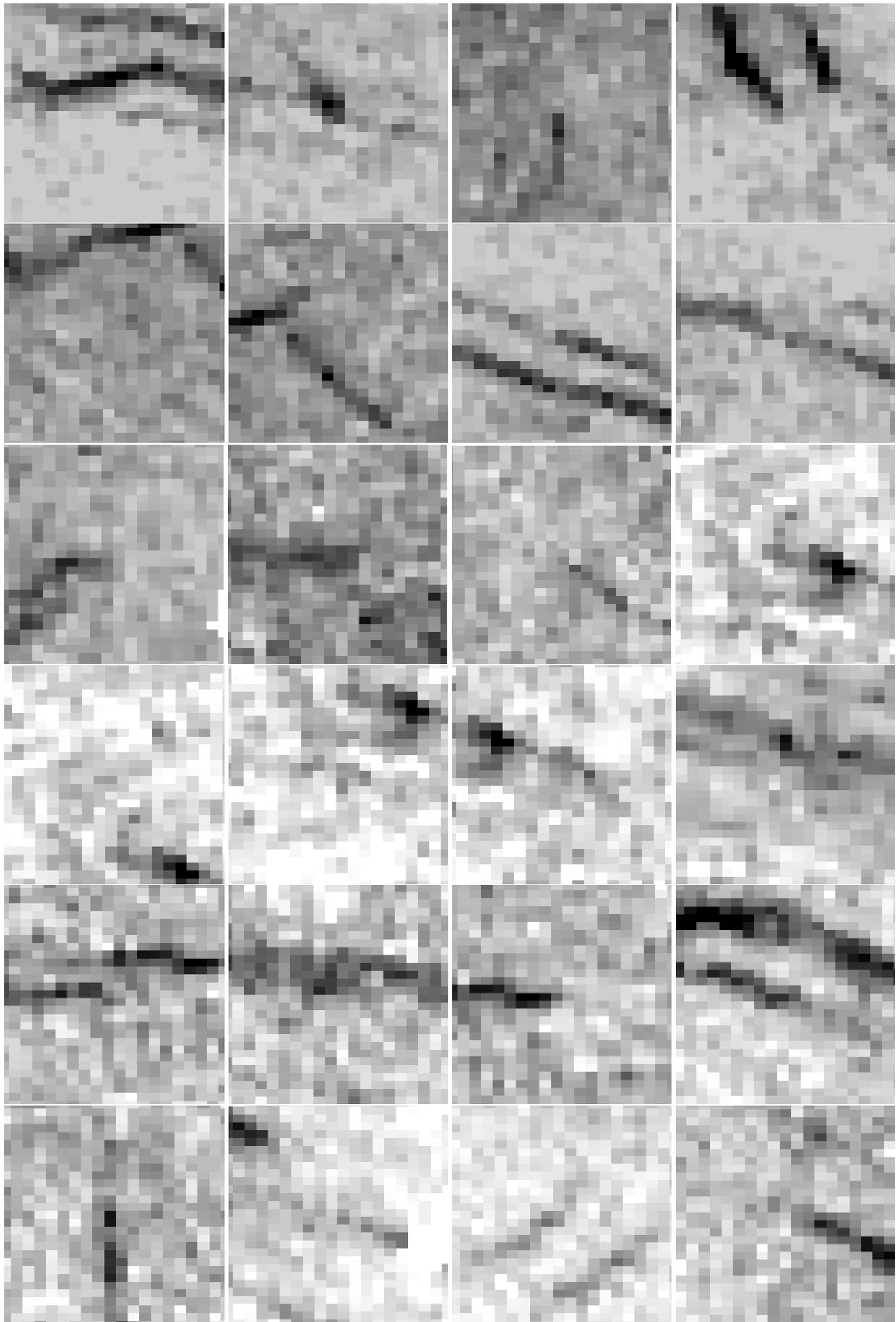
- [22] J.P. Veefkind et al. “TROPOMI on the ESA Sentinel-5 Precursor: A GMES mission for global observations of the atmospheric composition for climate, air quality and ozone layer applications”. In: *Remote Sensing of Environment* 120 (2012). The Sentinel Missions - New Opportunities for Science, pp. 70–83. ISSN: 0034-4257. DOI: <https://doi.org/10.1016/j.rse.2011.09.027>. URL: <https://www.sciencedirect.com/science/article/pii/S0034425712000661>.
- [23] J. van Geffen et al. “S5P TROPOMI NO<sub>2</sub> slant column retrieval: method, stability, uncertainties and comparisons with OMI”. In: *Atmospheric Measurement Techniques* 13.3 (2020), pp. 1315–1335. DOI: [10.5194/amt-13-1315-2020](https://doi.org/10.5194/amt-13-1315-2020). URL: <https://amt.copernicus.org/articles/13/1315/2020/>.
- [24] Justin Harper. *Suez blockage is holding up \$9.6bn of goods a day*. Mar. 2021. URL: <https://www.bbc.com/news/business-56533250>.
- [25] Luc Anselin. “Local Indicators of Spatial Association—LISA”. In: *Geographical Analysis* 27.2 (1995), pp. 93–115. DOI: <https://doi.org/10.1111/j.1538-4632.1995.tb00338.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1538-4632.1995.tb00338.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1538-4632.1995.tb00338.x>.

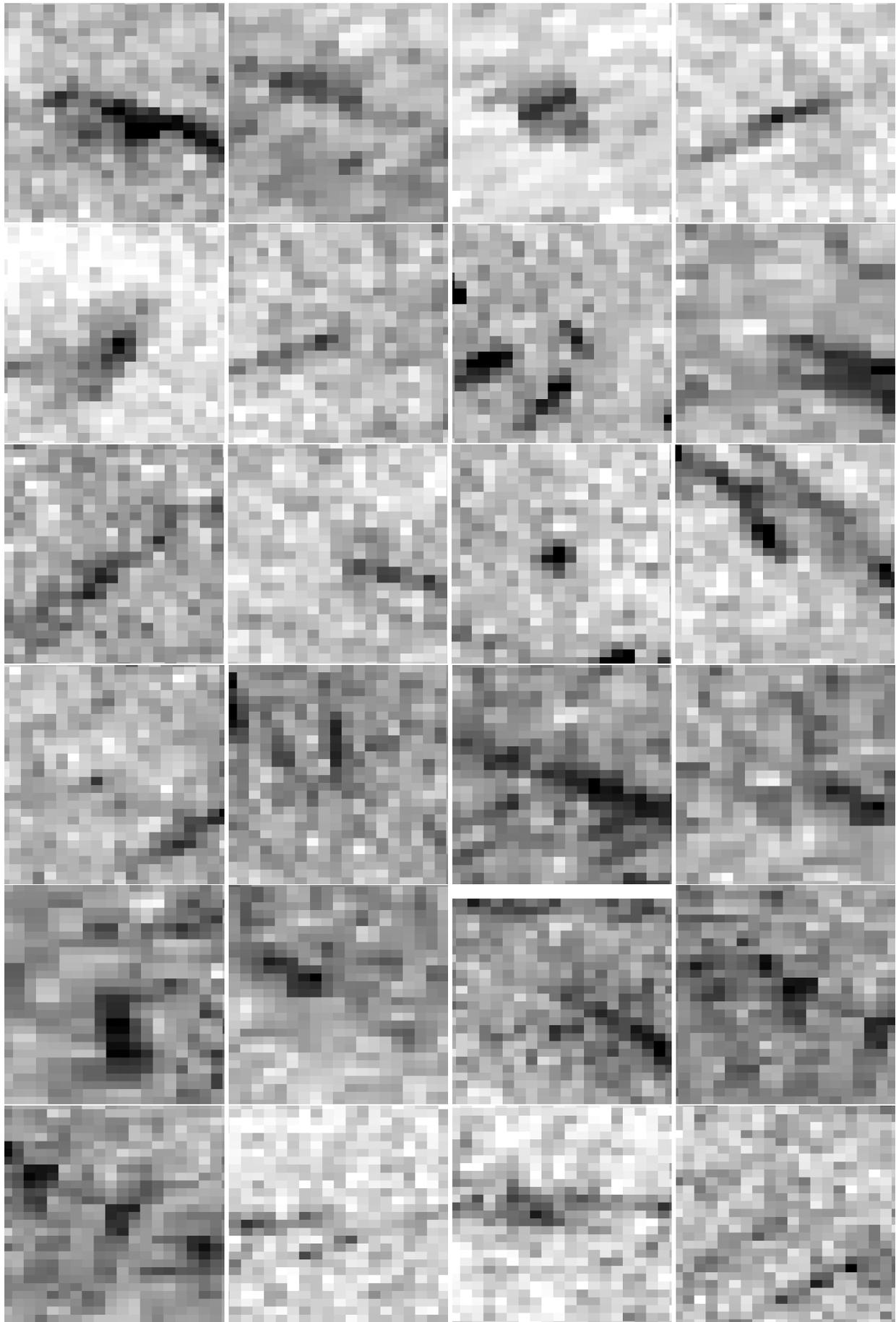
## A Appendix

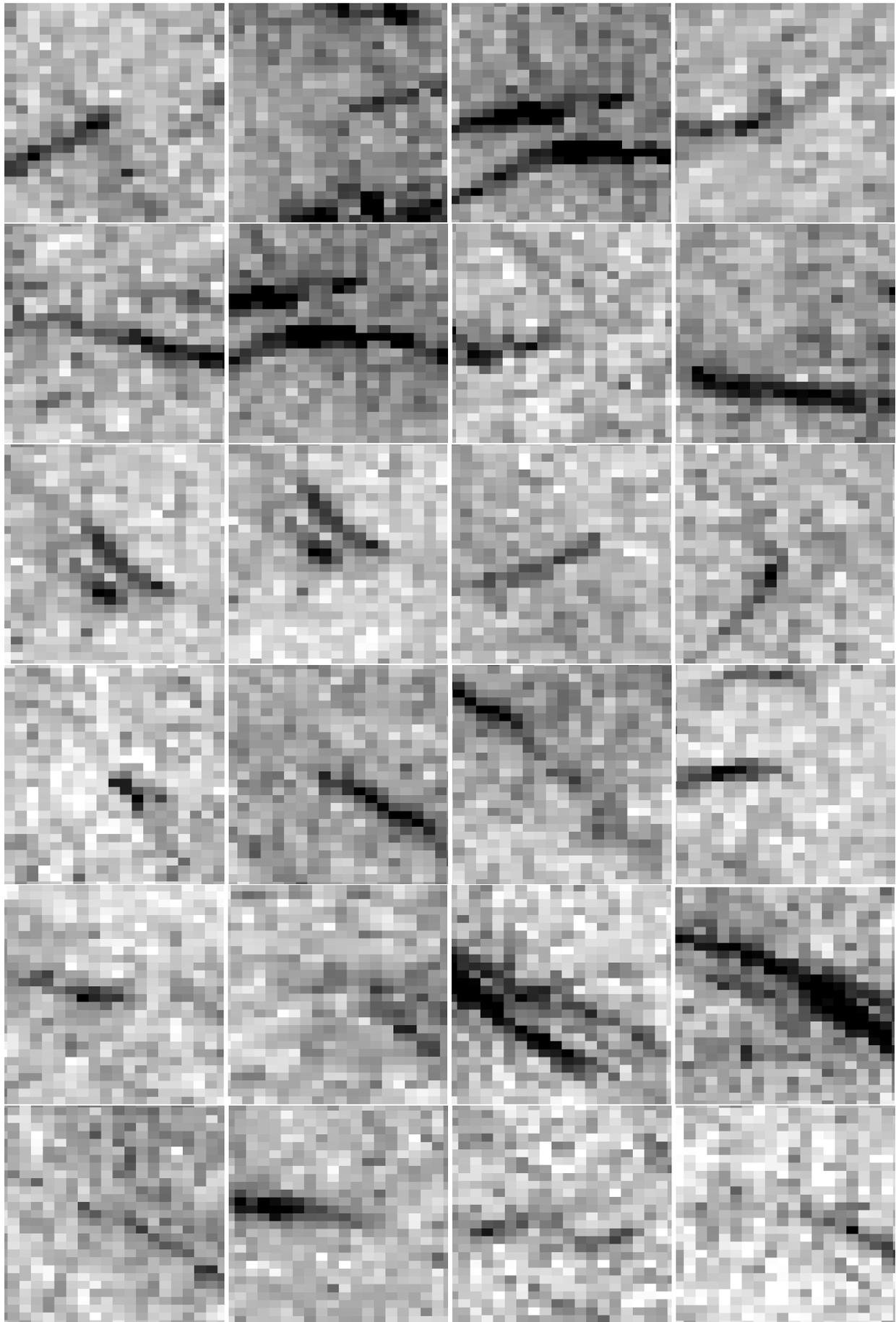
### A.1 An example of the test set labelled ”plume”

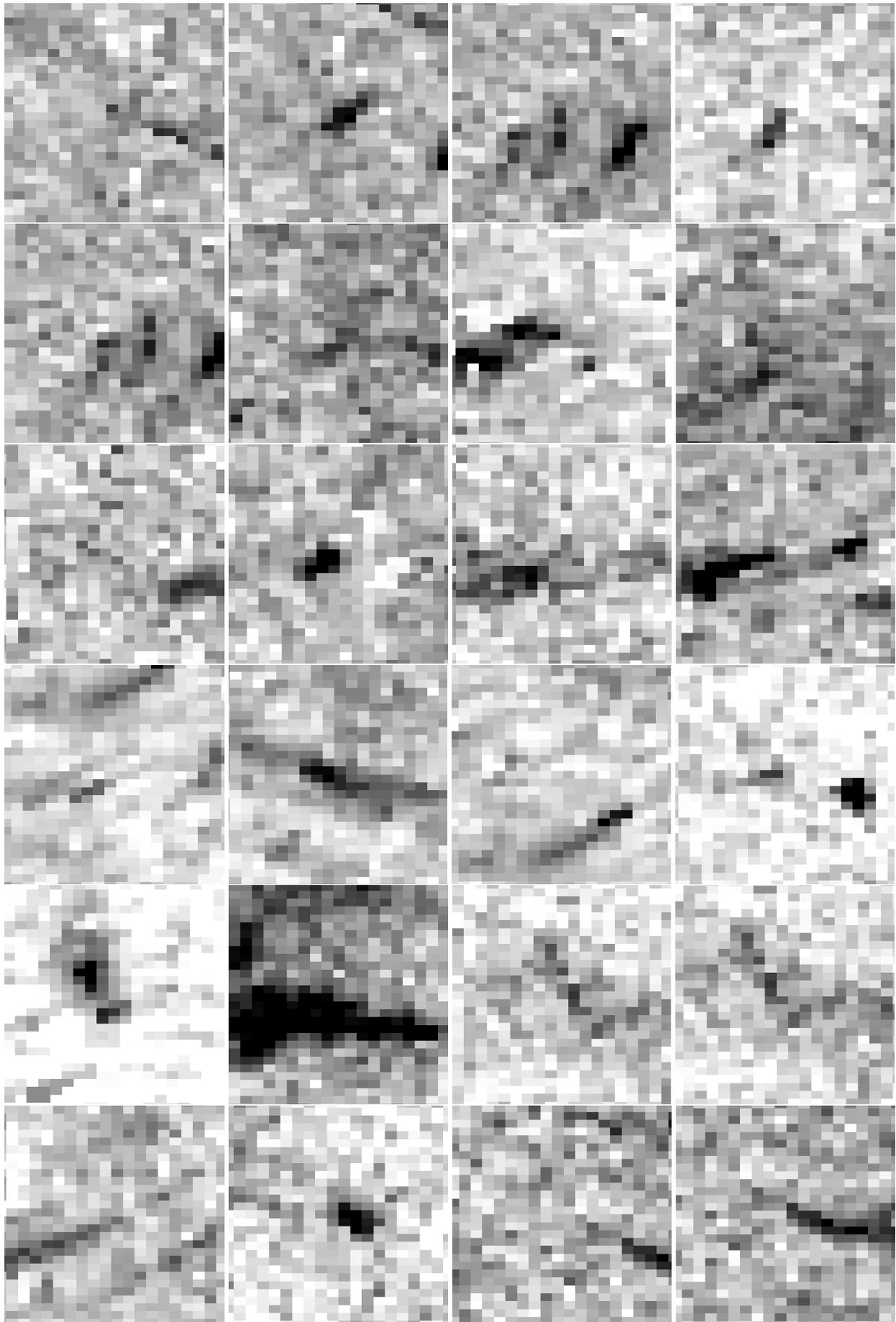












## A.2 An example of the test set labelled "no plume"

