JGR Space Physics

RESEARCH ARTICLE
10.1029/2021JA029683

Transfer Learning Aurora Image Classification and Magnetic Disturbance Evaluation

P. Sado[1], L. B. N. Clausen[1], W. J. Miloch[1], and H. Nickisch[2]

[1]Department of Physics, University of Oslo, Oslo, Norway, [2]Philips Research, Hamburg, Germany

Key Points:

- A pretrained feature extractor and a subsequent classifier can successfully detect aurora in all-sky images
- A validation on unknown, partially manually categorized images achieved a classification accuracy of 91%
- We predict physical quantities such as magnetic disturbance and cloud height from the underlying image feature representation

Correspondence to:
P. Sado,
pascal.sado@fys.uio.no

Citation:
Sado, P., Clausen, L. B. N., Miloch, W. J., & Nickisch, H. (2022). Transfer learning aurora image classification and magnetic disturbance evaluation. *Journal of Geophysical Research: Space Physics*, *127*, e2021JA029683. https://doi.org/10.1029/2021JA029683

Received 23 JUN 2021
Accepted 6 DEC 2021

Abstract We develop an open source algorithm to apply **T**ransfer learning to **A**urora image classification and **M**agnetic disturbance **E**valuation (TAME). For this purpose, we evaluate the performance of 80 pretrained neural networks using the Oslo Auroral THEMIS (OATH) data set of all-sky images, both in terms of runtime and their features' predictive capability. From the features extracted by the best network, we retrain the last neural network layer using the Support Vector Machine (SVM) algorithm to distinguish between the labels "arc," "diffuse," "discrete," "cloud," "moon" and "clear sky/ no aurora". This transfer learning approach yields 73% accuracy in the six classes; if we aggregate the 3 auroral and 3 non-aurora classes, we achieve up to 91% accuracy. We apply our classifier to a new dataset of 550,000 images and evaluate the classifier based on these previously unseen images. To show the potential usefulness of our feature extractor and classifier, we investigate two test cases: First, we compare our predictions for the "cloudy" images to meteorological data and second we train a linear ridge model to predict perturbations in Earth's locally measured magnetic field. We demonstrate that the classifier can be used as a filter to remove cloudy images from datasets and that the extracted features allow to predict magnetometer measurements. All procedures and algorithms used in this study are publicly available, and the code and classifier are provided, which opens possibility for large scale studies of all-sky images.

Plain Language Summary In the interest of auroral research and space physics, many images capturing the night sky have been taken automatically over the last decades. Sifting through these images manually takes a lot of time and is generally impractical. We use Convolutional Neural Networks (CNN), which are good at image classification to extract a set of numbers per image ("features") that capture the essential contents of the image. A Support Vector Machine (SVM) is trained to interpret these features and assign labels to the images. We search for the best configuration between different CNNs and SVMs and achieve up to 91% accuracy. To show that our method can be extended to other datasets, we classify half a million images from a different dataset and evaluate the performance of our classifier based on these results. We show that our classifier also excels at detecting clouds in images. It can therefore be used to filter unusable images from this kind of datasets. Based on the images' features, we create a model to predict disturbances in the Earth's local magnetic field. To enable other researches to work with our results, we use industry-standard, open-source software and make our algorithms and results available the same way.

1. Introduction

Created by particles precipitating into the ionosphere, the aurora are a direct consequence of interactions between the solar wind and the magnetosphere. A connection between different auroral shapes and behaviors simultaneously across the polar region has been first shown in 1964, when analyzing auroral images taken across northern continental America and Antarctica (Akasofu, 1964). The established model of auroral substorms, describes the activity of the aurora from its beginning, when the aurora is calm, over active phases toward another calm phase. This model was later refined to include the three currently used phases "growth", "expansion" and "recovery" (McPherron et al., 1973). Interaction between the solar wind and the interplanetary magnetic field lead to storage of energy in the magnetosphere during the growth phase. In the expansion phase energy is released, before the magnetosphere returns to normal conditions in the recovery phase.

Since the beginning of auroral research, images have therefore been an important tool to analyze and diagnose the complex processes in the ionosphere and magnetosphere, often supplemented by magnetometers measuring the local Earth's magnetic field or satellites performing similar measurements. All Sky Imagers, taking pictures of the night sky and capturing aurora in regular intervals, can for example, be found in Ny-Ålesund and Longyearbyen

on Svalbard, operated by the University of Oslo (UiO), in Canada and Alaska as part of THEMIS operated by NASA (Mende et al., 2009) and at the Yellow River Station in Ny-Ålesund operated by the Polar Research Institute of China.

Since their establishment in 1997, the imagers operated by the University of Oslo (UiO) have taken approximately 8 million images. We estimate that THEMIS is worldwide the largest producer of images. Taking one image every 3 s on 20 different imagers produces 24,000 images every hour. If a season has 12 weeks of days with 12 hr operation time each, 24 million images are taken each season. Analyzing and labeling images manually and consistently would take many humans to even keep up with the production of images, which is why researchers usually select smaller events on the timescale of a few hours or days and manually analyze the images in combination with other sources (see e.g., Murphy et al., 2013; Rae et al., 2017). Large-scale image data analysis requires automation to select the images.

First attempts have been performed by Syrjäsuo and Pulkkinen (1999) who determined the topological skeletons of the shape of aurora in auroral images which they later used to find images showing auroral arcs (Syrjäsuo et al., 2000; Syrjäsuo et al., 2001). Later Syrjäsuo and Donovan (2002) proposed an algorithm utilizing a kNN-classifier based on the images' mean and maximum brightness to differentiate between images showing aurora and images not showing aurora for which they reported 92% accuracy. The classifier has been improved over the years (Syrjäsuo et al., 2002; Syrjäsuo & Donovan, 2004) for example, by using Fourier descriptors on a segmented image (Syrjäsuo et al., 2004). This allowed them to obtain a rotation invariant descriptor of the auroral features present in the image and building a database upon which similar images could be identified and queried for. Further improvements (Syrjäsuo & Donovan, 2005) led to an automated classifier (Syrjäsuo et al., 2007) making use of Basic Gray Level Aura Matrices (BGLAM) to classify images into six classes (unknown, cloudy and four types of aurora) with 70%–80% accuracy.

Another approach is to use convolutional neural networks to categorize the images. A common benchmark for convolutional neural networks is the dataset provided for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015), which provides over one million labeled images in 1,000 classes. This database can be used to train networks without the costly process of having to label large amounts of training images. The 2012 winner of ILSVRC was the AlexNet convolutional network, which was the first to demonstrate GPU-based computing to train the network and achieved a top-5 error of 15.3%, compared to the second best entry of 26.2% (Krizhevsky et al., 2017). This was an important step forward, since the GPU computation allowed for much faster training of networks. When the original challenge retired in 2017 to focus on other objects, the top-performing network SENet achieved a top-5 error below 5% (Hu et al., 2020). These networks are made available publicly in their trained form and allow others to be used for validation of their results, comparison and further research.

Using the AlexNet architecture, Yang et al. (2018) finetuned the pretrained network with auroral images, taking local, regional and global features in the image into account and provide an algorithm that retrieves images similar to that one queried with an accuracy of 65%–70%. This was further improved by detecting regions of interest aligned along the magnetic field first (Yang, Wang, et al., 2019). To extract key local structures around the aurora also allows for retrieval of similar aurora images with an accuracy of up to 92% (Yang, Tao, et al., 2019).

Kvammen et al. (2020) used results obtained by Clausen and Nickisch (2018) to remove non-auroral images from their data and trained different established neural network architectures on the seven auroral labels they introduced earlier (McKay & Kvammen, 2020). The performance of their neural networks is compared to a support vector machine (SVM) and k-nearest neighbor classification trained on the histogram of oriented gradients and the method by Clausen and Nickisch (2018). With the best performing network they report up to 92% precision and 90% recall and F1-score displaying the neural network's ability to recognize challenging classes.

While these methods become more and more accurate, the underlying models are getting more sophisticated and are therefore difficult to implement or require expensive training or finetuning. They often also require filtered input data such that only images of aurora are shown. A convolutional neural network (CNN) predicts an image class, by first extracting a numerical representation of the image–the image features–which it then uses to predict the class label. In transfer learning, these features are used to train a model with a similar objective to the original model at greatly reduced training cost. This method has already been shown to be promising by Clausen and

Nickisch (2018), who reported 82% accuracy distinguishing aurora images between 6 classes - 3 different classes of aurora and 3 different non-aurora classes - and 96% for distinguishing between aurora and no-aurora.

Using the same dataset, we propose an improvement to this technique, by evaluating different pretrained networks to select the one best suited for this task and use an SVM instead of ridge regression for training the last classification layer of the network. We achieve 91% aggregated classification accuracy when distinguishing between images that show aurora and images that do not show aurora and 73% accuracy in the same 6 classes as the previous publication. We also show that the features extracted by the neural network can be used to infer information about the local magnetic field. The data supporting our **T**ransfer learning **A**urora image classification and **M**agnetic disturbance **E**valuation (TAME) is made available under the Attribution 4.0 International (CC BY 4.0) license (https://creativecommons.org/licenses/by/4.0/) and can be used to classify vast amounts of all sky images in a short amount of time.

2. Description of Data Sources

The first of our three data sources is an All Sky Imager, taking images of the sky when the Sun is below −9° elevation in intervals of 10–60 s in wavelengths of 5,577 Å and 6,300 Å, located in Ny-Ålesund near Sverdrup Research Station (78.92°N 11.93°E). The imager uses an electron multiplying CCD camera (EMCCD) with a filter with a spectral bandwidth of ≈2 nm allowing only the desired wavelengths to pass, taking images through a fisheye lense with a field of view of 180°. The camera produces images at a resolution of 460 px by 460 px which are stored as 16-bit gray level files. In order to protect the camera from saturation it is turned off when the Sun or Moon are within the field of view.

Second, we use data from a magnetometer in Ny-Ålesund operated by Tromsø Geophysical Observatory (TGO) https://geo.phys.uit.no/, 2021, obtained through IMAGE (Tanskanen, 2009). The magnetometer is located in close proximity to the all sky imager and takes measurements of the earth's local magnetic field X-, Y-, and Z-component. We use data with 60 s resolution.

Third, we use data from a ceilometer operated by AWIPEV Research Base at the Ny-Ålesund Research Station (Maturilli & Herber, 2017) that are described in Maturilli and Ebell (2018). This instrument measures the base height of clouds every 60 s. While this only gives the measure of cloud base height (CBH) right above the instrument, it is expected to be a good indicator of the cloudiness of the sky for a given time interval.

Although there are all sky images available since 2006, we will restrict ourselves in this analysis to the season of Nov 2010 to Feb 2011, because this is the year when the ceilometer reports the highest chance to have clear skies at any given time. This restriction still covers magnetic events of down to −800 nT, which is low enough to cover most major events. When we will later predict disturbances in the magnetometer measurements, we use the mean off all values in this timeframe as a baseline. For a timeframe of 4 months we can use this as a baseline, data covering several seasons would need more preprocessing. This could introduce unwanted or unforeseen biases into the data.

The ceilometer has been replaced several times over the years of operation, as well as the cameras taking the images. The cameras have been calibrated, ensuring that the measured brightness is the same with different cameras within known margins of error. The data obtained from the ceilometer may however be subject to inhomogeneities due to the replacements and long-term studies based purely on these measurements are discouraged (Maturilli & Ebell, 2018). Using only a timeframe of 4 months we do not need to account for these potential problems when validating the classifier.

In our analysis of the predictors in Section 4 we will analyze the representativeness of our data in its own context. If this method would be expanded to more years, a more in-depth analysis of the representativeness of the sample would be required. Information on how to explore the representativeness of a dataset for machine learning can for example, be found in McGranaghan et al. (2021).

Table 1
The Different Class Labels, Their Explanations and Machine-Returned Labels

Label	Explanation	Class 2	Class 6
arc	The image shows aurora with well defined edges with on or multiple bands spanning all or most of the field of view of the imager	0	0
diffuse	The image shows aurora that are fuzzy or patchy and do not follow a certain shape. The brightness is usually lower and of the order of the stars. This category is also often referred to as "patchy" in other publications.		1
discrete	The Image shows discrete, but not arc-like aurora. This could for example be swirls or crossing arcs.		2
cloud	The Image shows clouds or the dome of the imager is covered in snow.	1	3
moon	The image shows the moon but no aurora		4
clear sky no aurora	The sky is clear and no aurora are visible.		5

3. Methods

3.1. Preprocessing Images

Our images are scaled linearly between the 0.5% and 99.5% intensity percentile. Values below or above are set equal to the boundary of the percentile. The lower boundary ensures a more even background, the upper boundary removes very bright outliers not more than a few pixels large like bright stars or meteors but does not change bigger objects like the moon.

After scaling, we resize the images using bilinear interpolation to reduce the size of the images to the required input size of the neural network.

The images are then transformed by mean and variance as per the documentation of the neural network.

3.2. Image Classes

Table 1 shows the different classes, their labels and explanations. We differentiate between the three aurora classes "arc," "diffuse," "discrete" and three non aurora classes "cloud," "moon" and "clear sky".

Arcs are defined by bright bands of aurora spanning the whole image in a continuous line. These bands are well-defined with sharp edges, homogeneously bright and usually appear in the east-west direction. If multiple arcs are present, they are parallel and do not cross over each other.

Diffuse aurora are fuzzy or patchy. There is no clear structure or shape in the aurora. They are not as bright as the arc or discrete aurora.

Discrete aurora are similar to arcs in that they show bright features with well defined edges, but discrete aurora allow for inhomogeneous brightness, discontinuous bows, swirls or crossing features.

The non aurora classes show different features of the night sky when no aurora are present. The "cloudy" class covers a night sky covered in clouds, the "moon" class images where the moon is visible and the "clear sky" class when there is a clear night sky.

The color scheme we have used to label the classes in this table is the same used throughout the whole publication.

Several classes might be present in the same image for example, a partially cloudy image on an otherwise clear night sky. Images in the training dataset are however always assigned to a single class only corresponding to the class with the highest probability. The classifier will at a later stage return probabilities for each class, which we will discuss and show use-cases of in Sections 4.2 and 4.5.

3.3. Feature Extraction

To evaluate the images, we make use of the ShuffleNet V2 neural network pretrained on imagenet to extract numerical features from the images (Ma et al., 2018). The ImageNet database contains labeled sample images of different categories (e.g., animals, household objects, vehicles), which the neural network is trained to detect and classify such as to be able to classify a previously unknown image into one of the trained categories.

Neural networks originally trained for classification of images on a large set of image data like ImageNet do not have the ability to classify all sky images by themselves. The weights adjusted in the learning process however are tuned such that intrinsic image features like edges, basic shapes or patterns are recognized and receive a numerical description independent of the input image. For our purposes, we remove the last layer of the neural network - the classification layer - and are left with a 1,000 dimensional feature vector $\vec{\varphi}_i = \vec{\varphi}(x_i) = (\varphi_{i,0}, \varphi_{i,1}, \varphi_{i,2}, \dots, \varphi_{i,999})$ for every image x_i, that contains features of the image as numerical values. Under the premise that these values are a descriptor of the image, we use these values for further processing.

This transfer learning approach uses the fact that the core representation of images for a neural network are always similar. Similar to how a child learns that a square with a triangle on top is a house, the classifier can learn that if it sees an image whose features are 50% of the feature representation of a cat and 50% of that of a spoon, then these features represent an image showing auroral arcs.

This transfer learning approach is necessary, because we rely on a dataset with only 5,824 labeled images. We therefore do not have enough data to train and validate a new neural network and have to rely on a pretrained network. Training a non-linear algorithm, like finetuning a neural network for classification, on such a small sample size could easily lead to overfitting on the training data. The network would only learn the given images by heart and not understand the differences between the images. Instead of finetuning an existing and pretrained neural network to our liking, we therefore expand on the approach brought forward by Clausen and Nickisch (2018). While they showed the validity of this approach as a proof of concept, we want to evaluate more tools and validate the classifier on unseen data, while keeping the same philosophy of simple and out-of-the-box solutions.

Compared to classic image processing methods, where features are extracted by deterministic algorithms instead of by neural networks, the features we use are more abstract and not directly connected to the original image any more.

3.4. Classification

Because the neural network's classification layer was pretrained on classes different to ours, we replace the last layer with a different, easy to train algorithm. The two methods we will be using in this last layer will be ridge classifiers and support vector machines (SVM) (Wang, 2005).

Ridge classifiers are a special case of Tikhonov regularization, where the parameters used to describe the classifier are regularized equally to prevent overfitting and thus yielding a better result when presenting the classifier with previously unknown data (Raschka, 2015). A linear predictor might optimize the following loss functions measuring the discrepancy between a binary target label $t \in \{0, 1\}$ or $t' \in \{\pm 1\}$ a predicted function value $f \in \mathbb{R}$

$$\begin{aligned} \ell_2(t, f) &= (t' - f)^2, \quad t' = 2t - 1 \\ \ell_h(t, f) &= \max(1 - t'f), \\ \ell_c(t, f) &= -t\log f - (1 - t)\log(1 - f) \end{aligned} \tag{1}$$

where ℓ_2 is the L2-norm used in ridge regression and classification, ℓ_h is the hinge loss used by SVMs and ℓ_c is the binary cross entropy used in neural networks and logistic regression. For ridge classification, the regularized objective function becomes

$$L = \frac{1}{N} \sum_{i=1}^{N} \ell_2(y_i, \vec{w}^T \vec{\varphi}_i) + \lambda \cdot ||\vec{w}||_2^2 \tag{2}$$

where N is the number of samples, y_i the target value ($\{-1, +1\}$ for two-class classification or a binary label for multi-class), $\vec{w}$ the vector of weights that is optimized to find the best classifier and $\vec{\varphi}_i$ the vector of features

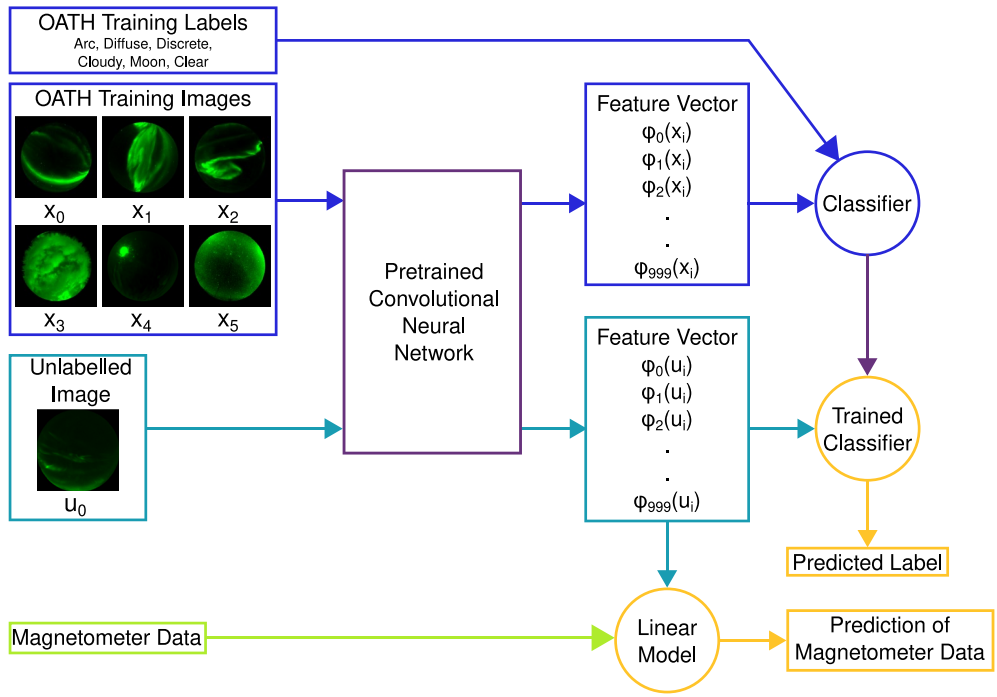

Figure 1. The flow of data (images and magnetometer) used in this work.

belonging to the image and training value. Depending on the regularization parameter λ, the additional term $\lambda \cdot ||\vec{w}||_2^2$ regularizes the magnitude the values of $\vec{w}$ can take. SVMs work by spanning the datapoints into an N-dimensional hyperspace divided by hyperplanes into as many regions as there are different labels (Wang, 2005). A new point is classified depending on which division of the hyperspace it falls into or might be rejected if it is too close to one of the hyperplanes.

SVMs employ a regularization parameter C substituting $\lambda = 1/(2\,CN)$ in Equation 2 to weigh misclassified training data. The larger C (the smaller λ), the more accurate the training data will be classified but the more vulnerable the SVM is to overfitting. If data cannot be separated by hyperplanes, a kernel can be used to transform the points into a higher dimensional space which can in turn be divided by hyperplanes. The radial basis function (RBF) kernel is one of the most commonly used kernels for this kind of application.

$$K(\vec{x}, \vec{x}') = \exp(-\gamma ||\vec{x} - \vec{x}'||^2) \tag{3}$$

The kernel measures the similarity between the two points $\vec{x}$ and $\vec{x}'$ where $1/\sqrt{2\gamma}$ corresponds to the effective length scale between two points up until which the kernel is effective. The larger the parameter $\gamma > 0$, the fewer points in the vicinity are taken into account. Ridge classifiers are easier to use and fine tune and SVMs require computation to train but are generally more precise. The ridge classifier is a linear classifier and can be optimized in closed form by solving a linear system, while the SVM requires a numerical approach to optimize its convex but non-linear hinge loss function $\ell_h(t, f)$ (Hastie et al., 2009).

We are aware that other loss functions like label regression could be applied, but followed the literature with binary cross entropy as a well established method. For noisy data the ℓ_2 norm will harshly penalize outliers, whereas binary cross entropy and hinge loss do not.

In Figure 1, we show the flow of how the images and other data are processed at a later stage. The training images (blue) are processed by the neural network (purple) to train a classifier (blue) using the known training labels. Previously unknown images (cyan) are preprocessed such that they are normalised to their maximum value the same way the training images are, before their features are extracted by the same neural network. These features can then be used to predict the images' labels (yellow) or predict for example, magnetometer data (green).

The OATH training images in the top corner ($X_0 - X_5$) also show examples of each class detailed in Table 1.

4. Results

4.1. Creating a New Classifier

To find the best pretrained neural network suitable for our work, we evaluated several network architectures available in PyTorch (an open source machine learning framework for Python, see Paszke et al., 2019) based on the accuracy of the prediction using the extracted features in a linear ridge classifier and the time it took to extract the feature vector. First using the network for feature extraction and then predicting the classes with a linear classifier or SVM using the extracted features effectively replaces the network's last layer that is used for classification with a different algorithm. This approach allows us to extract the images' features quickly on GPU, store them and resume processing on CPU. The process is easier to apply because feature extraction has to be done only once and training does not require expensive hardware. The extracted 1,000-dimensional feature vector is also more space efficient and therefore easier to access and use for later processing, for example, to predict other values from the images. Further improvements of a classifier like this one would also not require all features to be extracted again but could be run against the existing features, saving time. One terrabyte of images will be reduced to features the size of a few gigabytes, which can be loaded completely into the memory of an average desktop computer.

PyTorch provides its own library of pre-made neural network architectures, datasets and pretrained neural networks called "torchvision" (https://pytorch.org/vision/stable/index.html), which is usually installed alongside PyTorch. In addition to the pretrained neural networks provided through torchvision, we used pretrained networks provided by the python package "pretrained-models.pytorch" (Cadene, 2020).

The images we used were part of the OATH dataset as described by Clausen and Nickisch (2018). They were each normalised according to the value specified in the documentation provided with the pretrained model. Because neural networks in general require input images to exactly match their desired input size, we either zero-padded or resized our images using bilinear interpolation to match this requirement. Most neural networks we tested required an input size of 224 px × 224 px, about half the size of our original images of 460 px × 460 px. We provide a full list of all tested neural networks including the used parameters in the appendix in Table A1.

In Table 2 we show an overview of some of the tested pretrained networks. Accuracy was calculated once for two class labels ("aurora" and "no aurora") and for six class labels ("arc", "diffuse", "discrete", "cloudy", "moon", "clear sky/no aurora"). For this, we trained a ridge classifier on the features extracted by each neural network. The categories, their labels and explanations are shown in Table 1.

For each set of extracted features, the data was split into 10 sets of 70% (4,077 images total) training and 30% (1,747 images total) testing data each. The mean accuracy and standard deviation of the 10 trained and predicted runs is given as the accuracy and standard deviation in the table.

Ideally one wants much information in the training set to ensure the best classifier possible and much information in the testing set to accurately measure the performance. A balance between these two has to be achieved. For datasets in the millions like ImageNet (Russakovsky et al., 2015) commonly 5%–10% of the data are used for testing, for smaller datasets like ours (cf. Kvammen et al., 2020) 20%–33% of data are commonly used for testing.

Because the training images have been taken as time series with images usually 1 min apart, most images succeed images nearly identical to themselves. In a randomly split set of training and testing data, images in the testing set will therefore have an almost identical counterpart in the training data. This could allow bleeding of the training data into the testing data and a skewed representation of the testing accuracy. Instead of being split randomly, our testing data for cross-validation are 10 sets of consecutive images, each equal to 30% of the whole dataset, beginning at each 10%-marker. That is, the first testing set is equal to the first 30% of images, the second set of images from 10% to 40%, etc. After we have selected our hyperparameters using this method, we have chosen a split of consecutive images whose image classes are as close to the distribution of classes in the training class as possible.

Every neural network was given a ranking based on how the features extracted by the neural network performed in classification. The features extracted by the dpn68b for example, performed best in classifying the images in six classes and they were the 26th best features in classifying the images in 2 classes when using a Ridge Classifier. The table is sorted by the accuracy for six classes.

We can see, that for our method the features extracted by the top-performing neural networks show no significant difference in their predictive capability, neither in the 6 class accuracy, nor in the combined 2 class accuracy. As

Table 2
Time Used for Feature Extraction and Classification Performance Using a Ridge Classifier Based on the Features Extracted From the OATH Data by Several Neural Networks

Model name	Time (s)	Images processed per second	2 class acc (%)	6 class acc (%)	2 class acc ranking	6 class acc ranking
dpn68b[a] (Chen et al., 2017)	52.38	111	92.21 ± 1.53	72.83 ± 4.80	26	1
MNASNet 1_0[b] (Tan et al., 2019)	47.87	122	92.14 ± 2.25	72.39 ± 4.78	27	2
ResNeXt101_32x8d[b] (Hu et al., 2020)	76.84	76	92.44 ± 1.79	71.91 ± 5.08	23	3
PolyNet[a] (Zhang et al., 2017)	136.72	43	93.40 ± 1.31	71.70 ± 4.77	1	7
DenseNet161[a] (Cadene, 2020)	64.70	90	93.24 ± 1.60	71.26 ± 4.80	6	19
DenseNet161[b] (Huang et al., 2017)	60.50	96	93.24 ± 1.60	71.26 ± 4.80	5	20
ShuffleNet V2 x1_0[b] (Ma et al., 2018)	40.65	143	91.77 ± 2.50	70.93 ± 4.71	46	25
SE-ResNet101[a] (Hu et al., 2020)	57.30	102	93.30 ± 1.55	70.68 ± 4.55	2	29
MNASNet 0_5[b] (Tan et al., 2019)	42.29	138	91.80 ± 2.10	70.58 ± 4.24	45	32
SE-ResNet50[a] (Hu et al., 2020)	50.68	115	92.50 ± 1.70	70.30 ± 4.78	20	40
SE-ResNet152[a] (Hu et al., 2020)	67.71	86	91.99 ± 1.57	70.29 ± 5.13	31	41
InceptionV3[b] (Szegedy et al., 2016)	60.57	96	91.27 ± 2.04	69.94 ± 4.36	61	50
InceptionV3[a] (Szegedy et al., 2016)	59.16	98	91.27 ± 2.04	69.94 ± 4.36	60	51
VGG-19[b] (Simonyan & Zisserman, 2015)	56.42	103	90.72 ± 2.98	68.52 ± 5.06	69	71
InceptionV4[a](Szegedy et al., 2017)	73.14	80	90.44 ± 2.75	67.15 ± 4.95	72	78

Note. The last two columns show the ranking of the networks' features' performance for classification and the table is sorted by the ranking in the 6 class classification. Only a subset of the tested networks is shown. A full list can be found in the appendix in Table A1.
[a]Networks have been installed through "pretrained-models.pytorch" (Cadene, 2020). [b]Networks have been installed through "torchvision".

long as a current state of the art network is used, the extracted features carry equally good information for classification. In terms of accuracy, all networks are therefore equally suited for transfer learning for the purpose of classifying images. Therefore the classifier itself has to be improved, in order to obtain a significant improvement for classification.

We also see that the InceptionV4 model performed significantly worse than the results reported by Clausen and Nickisch (2018). While they achieved 96% for two classes and 85% for six classes, our implementation only reached 90% and 67% respectively. We attribute this to the fact that they used a random split of training and testing data, not accounting for the time-series nature of the data. Repeating their experiment with our method we achieve 90.07 ± 2.25 and 66.87 ± 3.87, which confirms this within the margin of error.

Another metric besides classification accuracy is the time taken for extraction. Here, "ShuffleNet V2" (Ma et al., 2018) performed best over all and it was able to extract features from all 5,824 images in 41 s. We also see fast performance for MNasNet with similar accuracies, because it was built with high performance especially on mobile devices in mind (Tan et al., 2019).

We then chose ShuffleNet for further processing because it was the fastest without performing worse in the classification task.

To improve on the classification, we evaluated SVMs with a linear kernel and an RBF kernel. Using tenfold cross-validation we perform a grid search for the optimal hyperparameters (See Appendix B for more information). We found that the performance of the SVM with RBF kernel was not better than the linear SVM. We have therefore chosen an SVM with linear kernel.

This also keeps the classification a linear problem. The non-linear RBF kernel could more easily overfit on our sample size.

For $C \approx 1.58 \cdot 10^{-4}$ we see the highest accuracy of 73.00 ± 5.22% for six-class and 91.27 ± 2.83% for two-class classification. The approach using cross-validation with several consecutive splits does not guarantee train- and test-splits with similar distributions of the different classes. For the final classifier we have therefore chosen a consecutive split in such a way that the distributions of classes in the training and testing split are

Table 3
Confusion Matrix for the Classifier Applied to the Final Hyperparameters and Testing Values

	arc	diffuse	discrete	cloud	moon	clear
arc	118	60	81	1	0	42
diffuse	31	222	44	1	0	6
discrete	25	57	314	0	0	8
cloud	0	40	18	146	16	20
moon	3	0	4	15	151	6
clear	9	32	28	7	1	240

Note. The rows contain the exact classes, the columns the predicted classes, that is, "arc\diffuse:60" means that 60 images showing "arcs" in the test set were predicted "diffuse".

as similar as possible (cf. Figure C1) To measure similarity, we have taken the Manhattan Norm between the different distributions as a measure. This returns a simple count of the differences between the distributions without giving different weights to classes that appear more often that others by using for example, the euclidean norm. Classifying the testing data we observe the following confusion matrix (Table 3):We see that the biggest confusion arises between the different classes of northern lights. The accuracy is 68.21%. If we aggregate this into two classes - aurora and no aurora - we achieve an accuracy of 89.00%. In Table 4 we have summarized some important metrics of the classifier. We can see that there are differences in the performance of the classifier for the different auroral classes. The recall for diffuse and discrete aurora with 0.73 and 0.78 is almost double the value of arcs. The precision of diffuse aurora is however also lower than the other auroral classes. In the confusion matrix we can see, that cloudy images are often mistaken for diffuse aurora but not the other way around. Similarly, arcs are often mistaken for discrete aurora. The cloud and moon classes both show high precision values, the misclassified images are also almost all in another non-auroral class. The classifier could therefore be a good tool to remove images with uninteresting features. This is supported by again aggregating the images into aurora and no-aurora, where we see high recall of aurora images and high precision for no-aurora images.

Another issue is that northern lights images are classified as images showing clear skies and vice versa. In Figure 2 we see an explanation why the classifier might have problems discerning between images showing clear skies and different types of aurora. Each row in the figure shows two almost identical images, where the left image has been assigned the "clear skies" label and the right image one of the different classes of aurora each. All of the images show a strong feature near the border of the image but an otherwise almost clear sky. Some of these images still labeled as "clear skies" already show aurora. Near the border of the image, aurora might also be cut-off and what is visible always appears as an arc, although the cut-off part of the aurora might require labeling as another class. Because so little of the aurora is visible and it is often distorted by the fish-eye lense near the edge of the image, it is safer to reject them. These borderline cases are already ambiguous when labeled by a human, because different people would draw the line between usable and not-usable somewhere else.

4.2. Evaluating Unknown Images Using the New Classifier

With the new classifier we are now able to classify any image taken with an all sky imager. On a dedicated computer with a GeForce GTX 1080 Ti, extracting features and applying the classifier to all 8 million images taken by UiO and available to us took less than a week, the bottleneck being the hard drive. We make the classifier and supplementing code available with this publication. Applying it to the THEMIS images could classify all images in about 3 months on our computer. On other computers, the speed of the application will vary depending on the available hardware.

Table 4
Summary of Metrics for the Classifier, Taking Different Classes Into Account

classes	precision	recall	f1-score	number of images in test set
arc	0.63	0.39	0.48	302
diffuse	0.54	0.73	0.62	304
discrete	0.64	0.78	0.70	404
cloud	0.86	0.61	0.71	240
moon	0.90	0.84	0.87	179
clear	0.75	0.76	0.75	317
Aurora	0.88	0.94	0.91	1010
No Aurora	0.91	0.82	0.86	736

We use the previously mentioned timeframe (cf. Section 2) to show some examples and test how well the classifier has performed. We obtain the distribution of classifications for the images as shown in Table 5. We can see that about 35% of the images are classified as showing some form of aurora, almost 50% are classified as cloudy and the remaining 15% classify a clear night sky. Only 26 images in total were classified as showing the moon, which is the first sign that the classifier works as designed, because the camera is turned off when the moon is visible to avoid over-saturation.

Out of the 198,438 images classified as aurora, less than 5% are classified as arcs, diffuse and discrete aurora are almost equal. This might either mean, that arcs are not prevalent in our dataset and that arcs are therefore overrep-

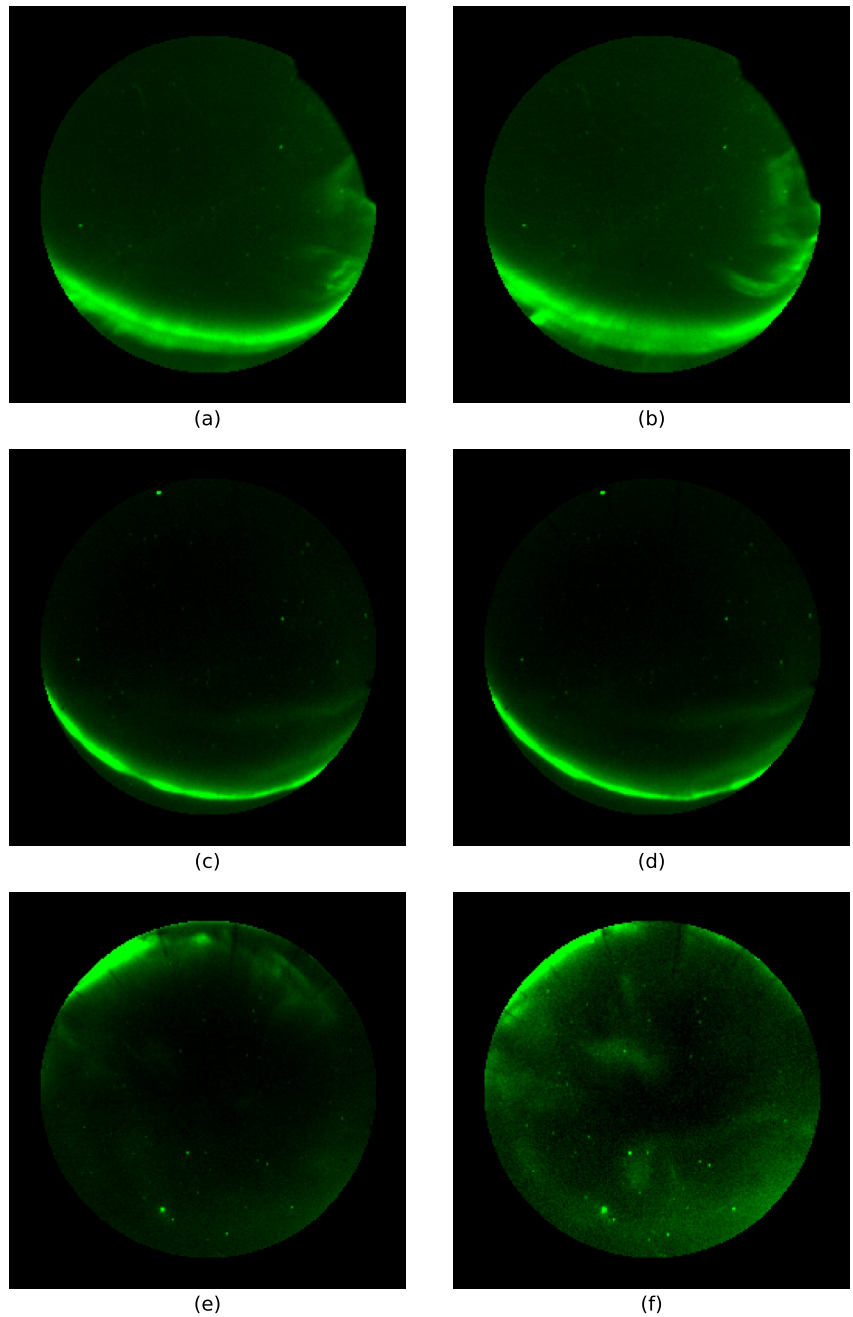

Figure 2. Labeled images of the OATH dataset. (a) Image 00868, (b) Image 00869, (c) Image 02052, (d) Image 02053, (e) Image 03955, (f) Image 03956. Images (a, c, e) are labeled "clear skies/ no aurora" each, Images (b, d, f) are labeled "discrete", "arc" and "diffuse" respectively.

resented in the training dataset or that our classifier failed at detecting and classifying arcs. Considering the low recall values for arcs on the testing data (cf. Tables 3 and 4), we think that this is a problem with the classifier, not our data.

Figure 3 shows the six images with the highest confidences in each class. The probability for each class is plotted as bar graph on the right-hand side of each individual image. The bars are normalized to the maximum value in each image and sum up to 1. The dashed line is drawn at a probability of 16.7%, which is the probability the classifier is correct by pure chance. The bars are ordered the same way we present the names, classes and corresponding colors in Table 1.

Table 5
Distribution of Predicted Classes in Images Taken in the 2010/11 Season

class	count
arc	9,200
diffuse	92,582
discrete	96,656
cloud	265,569
moon	26
clear	86,253
Aurora	198,438
No Aurora	351,848
Total	550,286

The images are shown after preprocessing as seen by the feature extractor. Although the resolution has decreased because the images have been reduced in size, the auroral features are still well distinguished.

In Figure 3f we can see that the images are grainy. Because we removed outliers in brightness and then scaled the image toward its minimum and maximum remaining brightness, if the minimum and maximum brightness are close to each other, small differences in brightness are more easily visible, giving the impression of a grainy image. The same is true in the other direction, meaning that a grainy image indicates an image with low variance in brightness.

We cannot trace back the original brightness of these images, but in general a homogeneously dark image is more likely than a homogeneously bright image, because there are no natural sources of light that would cause homogeneous lighting in times the camera is turned on.

The only exception could be a snow-covered dome, where the snow scatters the light to illuminate the whole dome evenly. In that case images would look similar to a pitch-black but clear night sky or homogeneous cloud coverage.

For the images classified as arcs (Figure 3a) we see different images of arcs. All images show one well-defined bow of auroral arcs against an otherwise clear and dark background. Arcs often follow an east-west direction but can also appear in north-south direction. In this demonstration we can only see east-west arcs, but the images used to train the classifier have been randomly rotated before extracting their features. We therefore assume that the classifier's bias toward rotation of the images has been minimized.

In Figure 3b we can see images classified “diffuse” with high probability. While there is some underlying structure remaining - for example, two bows going from left to right in the image in the middle right - there is no clear boundary between these structures and the background and much of the aurora is not a lot brighter than the background. The image in the middle left also shows some structure near its bottom, but there is again no clear boundary between the aurora and the background and the overall brightness is again low compared to the background.

Figure 3c shows the discrete, but not arc-like aurora. The top left image shows two bow-like structures. The brightness of these bows is uneven and in some places they break up, before continuing at the previous brightness-level. The upper structure also has a very bright patch near its top-right edge. The top right image is similar in makeup, with two bows in uneven brightness and a bright patch near one of the edges.

The images in the middle row show bows with uneven brightness. The structures of the images in the bottom row are larger than in the previous images but again uneven in brightness.

The characteristic feature of all images shown here, is that they show elongated, sometimes curly structures with well defined edges against a dark background. The brightness along the structure fluctuates down to almost background level at some points. These fluctuations and curls distinctively distinguish these structures from arcs.

Figure 3d corresponds to cloudy images. We can see that there was auroral activity in the background of some images, but in every case the aurora is obstructed by clouds.

Figure 3e was to label images as showing the moon. To avoid oversaturation, the camera is disabled when the moon is in the field of view of the imager. The whole dataset therefore does not contain a single image showing the moon and only 26 images out of the subset of 550,286 of the 2010/2011 season in total were labeled as such. What the classifier seems to have picked up on are images that show a clear background with single, very bright but small spots. In all 6 images presented here, the second highest class is for discrete aurora. These images fit the definition of discrete aurora, because the structures are bright against the background and show well defined, sharp edges. Because these misclassifications are so rare, they can safely be neglected.

Figure 3f shows a clear night sky with some very faint auroral activity near the bottom left field of view of the imager. These images would have likely been rejected by a human because they do not show enough and clear

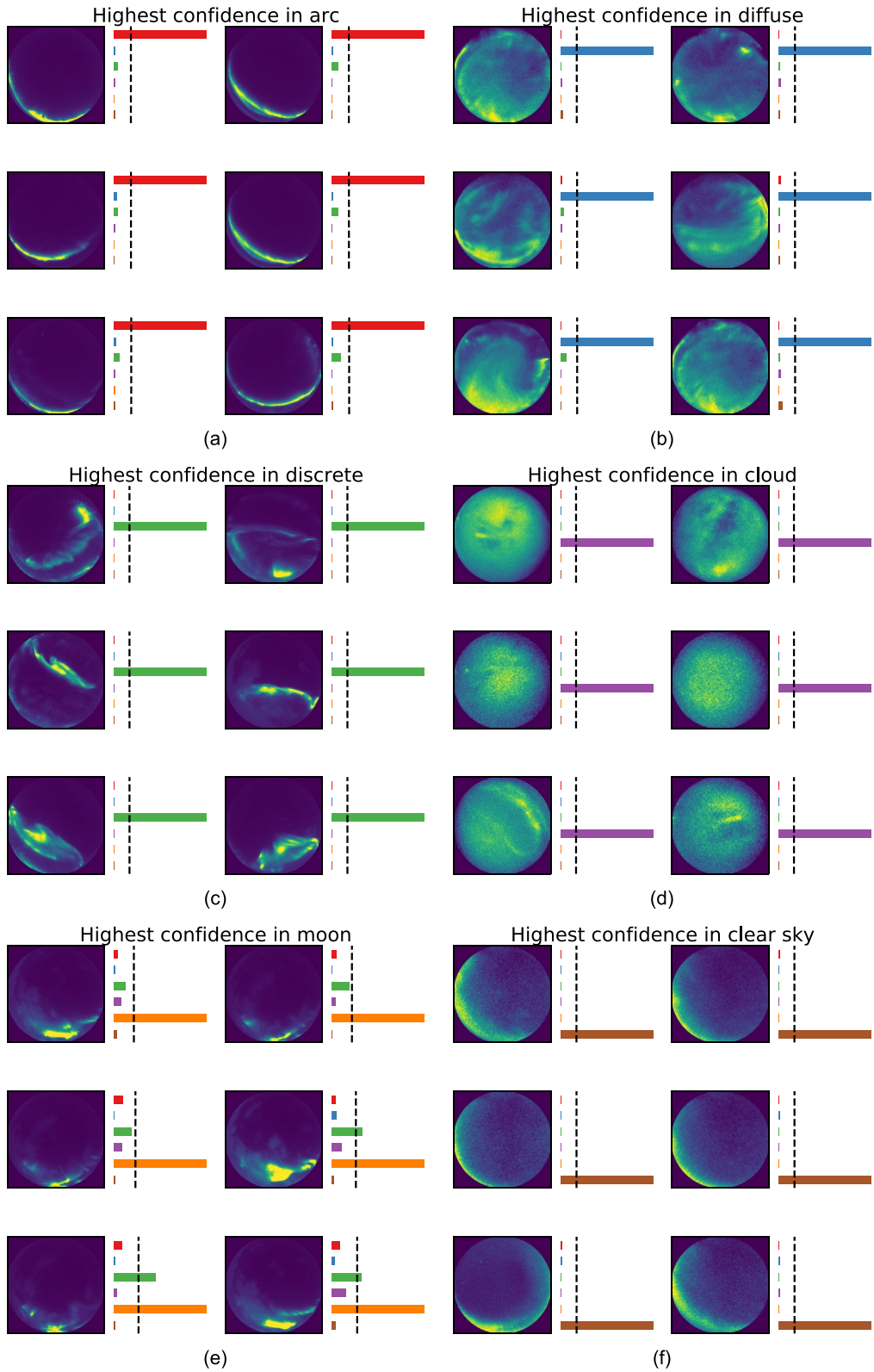

Figure 3. Images with highest confidences for each class (a) arcs, (b) diffuse aurora, (c) discrete aurora, (d) cloudy, (e) moon (f) clear sky. The probability assigned to each class are given as bar graph normalised to the highest probability for each image beside each image. The dashed lines indicate equal class probability of 16.7%. The colors of the bars correspond to the colors of the classes shown in Table 1. Images are shown after preprocessing as seen by the feature extractor.

aurora to be useful for classification. Aurora appearing near the edge of an image have some of their features cut off, therefore all auroral classes look the same or at least similar, when near the edge of an image.

The examples show auroral images that clearly belong in the classes they have been assigned by the classifier. There are some false positives for the "moon" class, but those images are rare and the classifier is not very confident in them. The classifier is therefore very confident in images that would have been labeled with the same confidence by a human operator.

The same way we analyzed the images the classifier had its highest confidence in, we can analyze the images with the lowest probability of being in each while still assigned that class. From every class we selected those images, where the image was assigned that class, but with the overall lowest probability of all images in that class. This is a measurement how confident the classifier was in this specific image and class relative to the other classes. We have plotted the images in Figure 4, the style is the same as for Figure 3.

For the three aurora classes and the clear sky class we see that there are mostly images with bright crescent shapes aligned with the edge of the image on an otherwise clear sky. These are similar to the edge cases in the training dataset we demonstrated above. For arcs the middle left and bottom right as well as the middle left image for the diffuse class deviate from this pattern. The aurora visible in these images are very small and can barely be made out against the background. Most likely images with faint or small aurora are underrepresented in the training set. The image labeled diffuse has been classified correctly, albeit with very little confidence.

For the cloudy and clear sky classes we see the same pattern as in most of the aurora images. For all of these images the probabilities of the different classes are very close. The last image in the bottom left is almost completely dark. The probability for the moon-class is very high in this image compared to previous examples. This might be, because the only time the classifier has seen completely dark images without any stars, clouds or aurora has only been when the moon was visible in the image. Dark images without any features are not present in the training dataset.

We would like to point out that this method of min-maxing - selecting the classes by their lowest probability where this probability was the largest compared to the remaining probabilities - is just one heuristic technique to explore situations where the classifier is unsure. Another method could be to look for images where the probabilities show high entropy. This should give similar results to the images shown in Figure 4, because the probabilities shown there are almost equally distributed.

At last we show examples of tiled images. These are given in Figure 5. After the original preprocessing, the images were tiled into 16 56×56 px sized squares. For the images in the left column (Figures 5a, 5c, 5e), these tiles were replaced by black tiles, thus removing all information previously present in this part of the image. For the images on the right (Figures 5b, 5d, 5f), the classification given below each tile corresponds to a classification based only on each specific tile. To do this, all other tiles but the tile to be classified were replaced by black tiles. Therefore only information present in a given tile was kept and used for feature extraction and classification. If the classification of the tile differs from that of the whole image, the class is written in bold letters to emphasise the change. In the second form of processing only 1/16 of the original image and information remains. The classifier has not been trained on images on this format and the results obtained from this have to be treated cautiously.

The images in the top row (Figures 5a and 5b) correspond to the last image in Figure 3e (high confidence in moon), the images in the middle row (Figures 5c and 5d) to the fourth image in Figure 3a (high confidence in arc) and the images in the bottom row (Figures 5e and 5f) to the third image in Figure 3c (high confidence in "discrete").

In Figure 5a we can see that most of the image can be removed and the classification as "moon" remains. Only when the second tile in the third row is removed, the classification changes to "discrete". This tile only shows a clear background with some clouds, but some information contained in this tile seem to be crucial for the classification. The classifier could have learned that the moon is often brighter than aurora and illuminates clouds in its proximity. Because we have replaced the tile with a black tile, it would correspond to a clear background, unusual for the classifier.

The features in the middle tiles in the bottom row in Figure 5b are what we previously attributed to cause the misclassification. Now we see that when isolating these features, the image is not classified as containing the

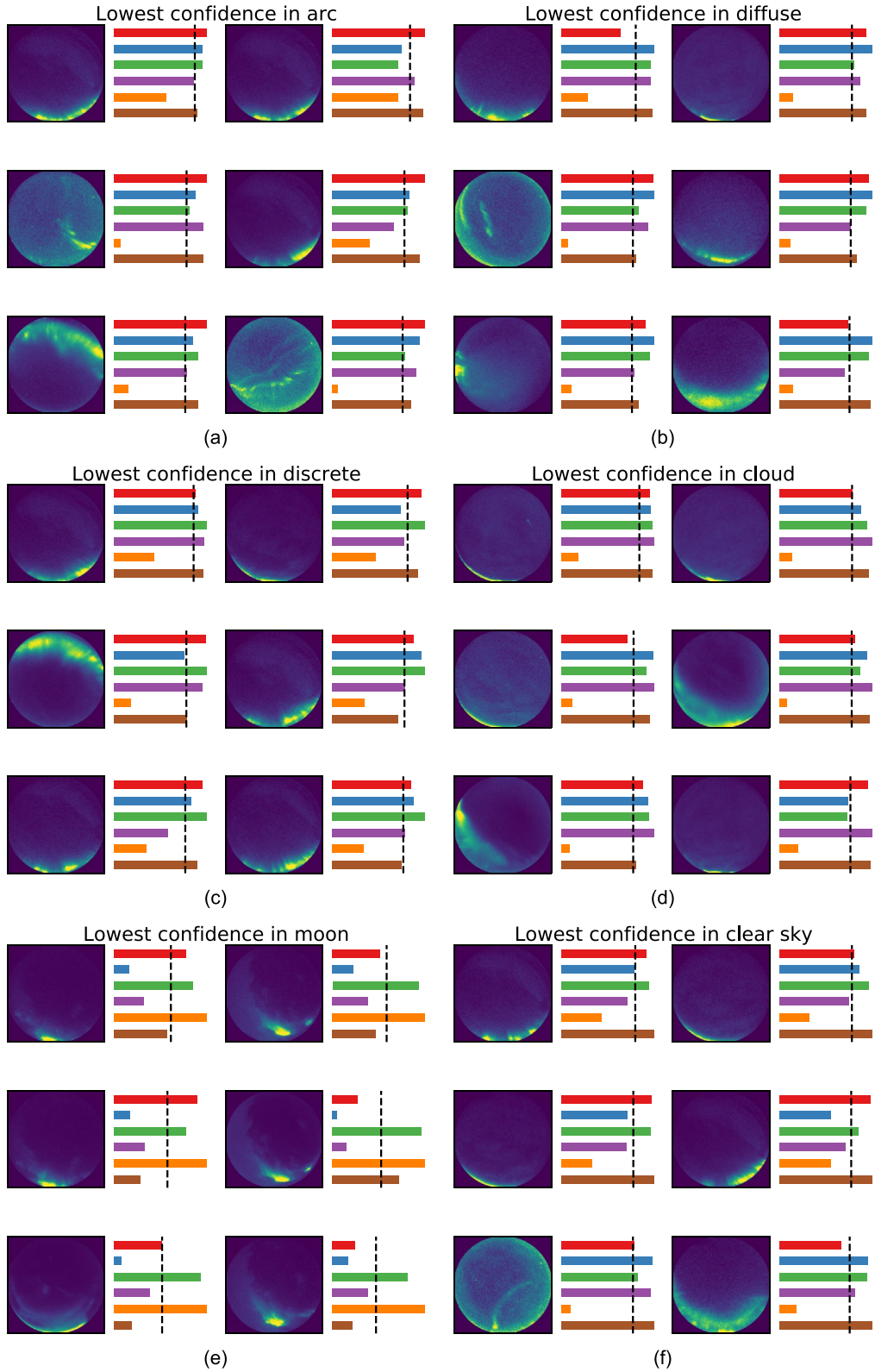

Figure 4. Images with lowest confidences for each class (a) class 0, (b) class 1, (c) class 2, (d) class 3, (e) class 4 (f) class 5. The probability assigned to the label by the classifier is given below each individual image. The dashed lines give a probability of 16.7%. The colors of the bars correspond to the colors of the classes shown in Table 1. Images are shown after preprocessing as seen by the feature extractor.

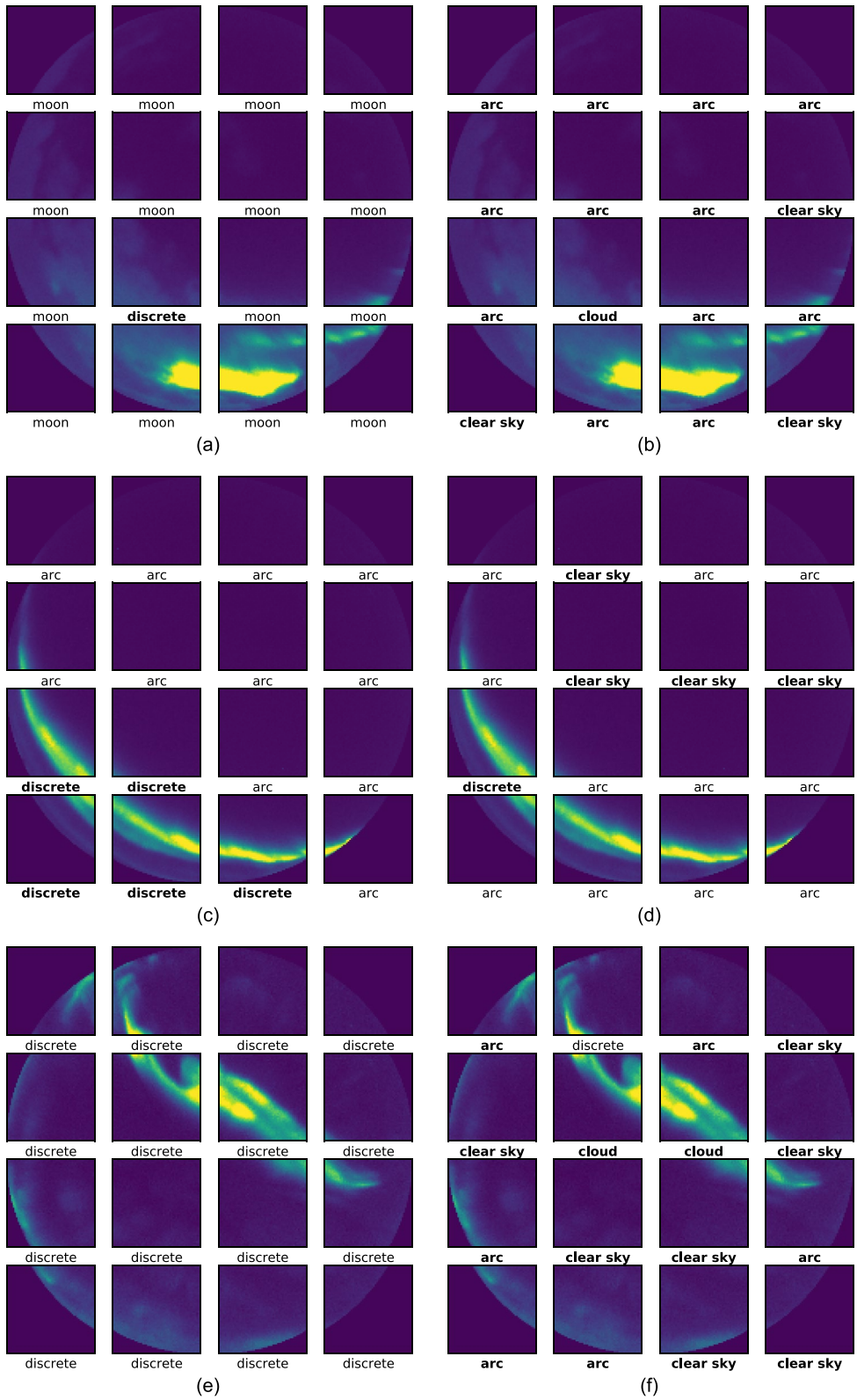

Figure 5. Classified images after tiling. Left column (a, c, e): the tiles were replaced by black tiles, for the right column (b, d, f), the classification is based on one tile only. The images in the top row correspond to the last image in Figure 3e; the images in the middle row to the 4th image in Figure 3a and the images in the bottom row to the 3rd image in Figure 3c. The respective predicted classes are printed below each tile, if the class is written in bold letters, the tiled classification differs from that of the original image.

moon, although the first tile matches the moon in size and brightness against the background when compared to the training images.

We conclude that more than just a bright object of roughly the same size is necessary for the classification. This could be a presence of clouds in the image, reflections of the moonlight on clouds or on the dome of the camera, or even lens flares, which can be found on some of the training images (cf. Figure C2). We also see that many of the single remaining tiles are classified as "arcs". Because we want to preserve scale, we have to replace all the remaining tiles with black ones instead of resizing the desired tile to full image size. This replacement may introduce sudden, large gradients on the edges of the tiles. Because arcs show well defined and sharp edges, the images will have large gradients in these parts. If the classifier has learned this behavior, this might be a reason why these tiles are misclassified as "arcs".

In Figure 5c we have an image previously labeled as "arc". We can see that we can safely remove the tiles that make up the background in the top right part of the image without changing classification. The first tile in the second row and last image in the fourth row are the tail of the arc and can also be removed without changing the classification. If we however remove any of the parts within the arcs and therefore break the arc apart, classification changes to "discrete". This suggests that the classifier has successfully learned that arcs and discrete aurora are similar, but that one important difference is that arcs need to be connected.

When observing tiles by themselves in Figure 5d we see again that some of the background tiles in the top right are classified as arcs, but also sometimes correctly classified as "clear sky". The arcs in the bottom part of the image are correctly classified but because we have seen how false positives are generated by this type of analysis, we cannot be certain that this analysis of padded tiles translates well to the performance of the classifier on whole images.

This image also illustrates well why we decided to resize our images, instead of cropping toward the center. Based on our original image size and the desired input size of the neural network, when cropping our image only a part about as large as the four center tiles would have remained. Any information about the aurora outside would have been lost.

At last we show the third image in Figure 3c which was previously classified as "discrete". No matter which tile is removed in Figure 5e, the image retains its correct classification. In the third tile of the second row, we remove about half of the visible aurora. This and Figure 5c suggest that the classifier is robust and works even if only small parts of the image show aurora, as long as the aurora follows a known shape. If we observe the classification based on the tiles in Figure 5f, we see that the classification in the parts that show aurora often does not hold. The top left and bottom right parts of the aurora are classified as diffuse aurora or arcs, whereas the main part in the middle is classified as cloudy. The two tiles directly below and one to the right have been classified correctly as "clear".

Again, we cannot be certain that the classifier picked up on information previously trained on, due to the way the padding affects the feature extraction. Another way would have been to again scale the image to the input size, which would have removed problems with detecting the background, but any information about the scale of the structures would have been lost.

In the previous we have demonstrated an improved way to classify all sky images by first extracting their features using a convolutional neural network before using these features in an SVM for classification. The classifier performs well on the testing data with 73.00% $\pm$ 5.22% six-class and 91.27% $\pm$ 2.83% two-class accuracy. While this number is lower than in previous publications, we found that the choice of training and testing data on this type of images is very important and the test-results need to be validated thoroughly. We are therefore confident, that there is little bias toward our training data and that our classifier will be able to handle unseen images from different sources with similar accuracy as the training and testing data.

When analyzing our classifier, we find that the biggest confusion for the classifier arises from images that are ambiguous to humans as well. These are often image with aurora near the edge of the image.

We also found that the classifier picks up on subtle, but important information, such as that arcs need to be connected or that images of the moon are not just a bright patch against a dark background but show reflections on the image.

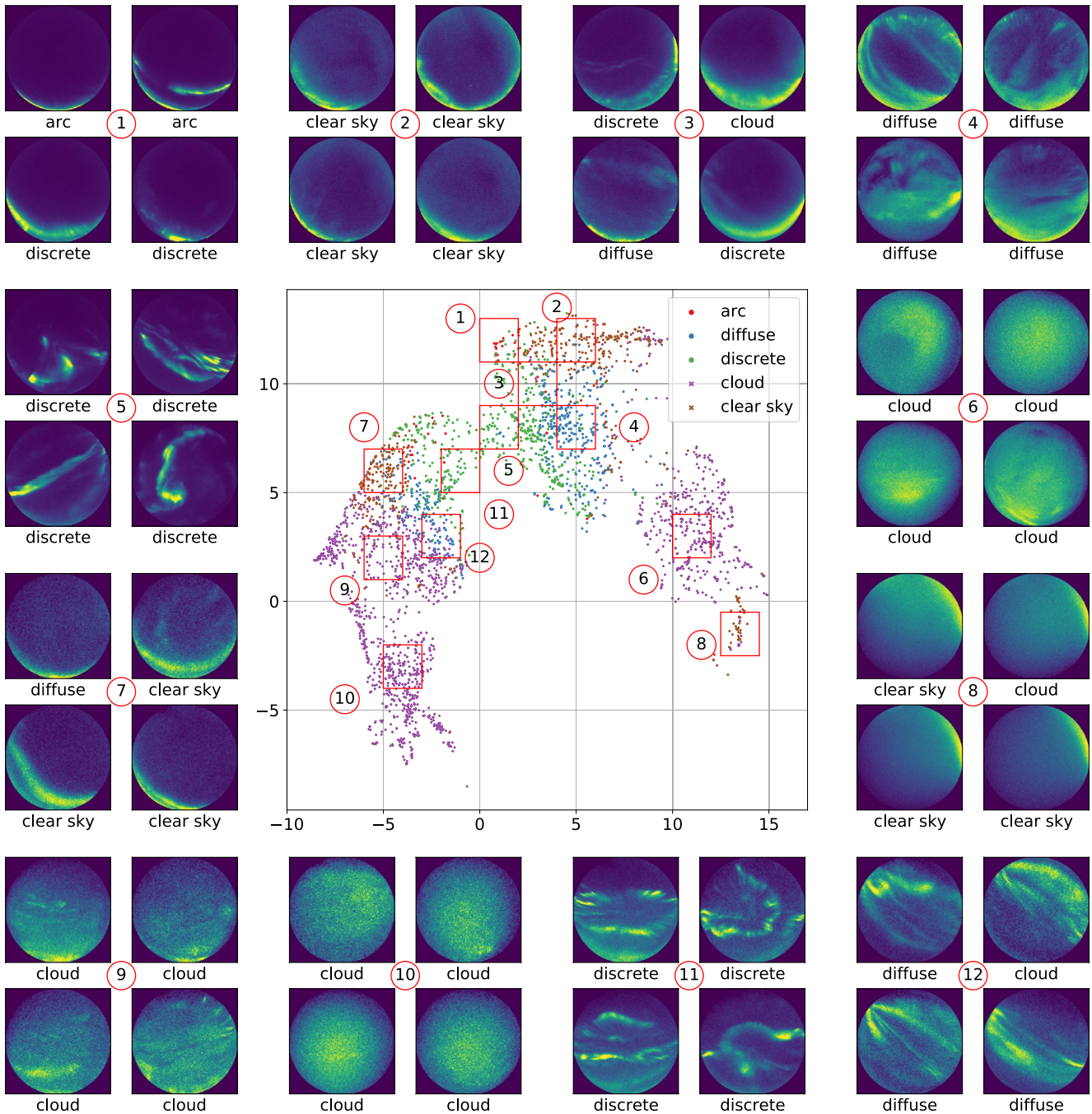

Figure 6. Features embedded into a 2D Space using the UMAP algorithm. Some areas of interest are highlighted by red boxes with four random images from within each area shown around the mapping. The numbers by the figures on the outside correspond to the numbers of areas marked in the 2D space. Images are shown after preprocessing as seen by the feature extractor.

4.3. UMAP of Features

Next, we want to understand how the extracted features relate to the images and their predicted classes. In the middle of Figure 6 we show how the features can be mapped into a 2D space using Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) (McInnes et al., 2018). We chose UMAP over T-SNE because this way global structure is preserved in the mapping. While we do not need this for further processing, this algorithm is good at picking out similarities and differences in numerical features and therefore allows us to display the 1,000-dimensional feature space condensed into a 2D space. All 550,286 images have been used to

create the map and a random selection of 3,000 images has been used to create the figure. Each point in the plot represents one of the 3,000 selected images. The colors represent the classes they belong to, using the same color scheme as explained in Table 1 and used in Figures 3 and 4.

The UMAP algorithm calculates a lower-dimensional feature representation of the original high-dimensional feature representation, grouping points of data with similar high-dimensional features close together and those with different features far apart. This means that points which lie close together in this mapping represent images which show similarities in their original features, images that are placed far apart are also far apart in their features. If the neural network succeeded in extracting the most important information out of the images into the features we expect to see visually similar images close to each other and visually different images to be far apart.

To illustrate this, we have selected some parts or clusters of the mapping and plotted some images contained in these areas. Those areas are marked by red rectangles in the mapping and assigned a number which connects them to one of the groups placed around the mapping. Each of the groups consists of four randomly selected images with the predicted class given below the image and the number assigned to the group in the middle between the images.

We see a butterfly-like shape, with images mostly labeled as aurora between the wings. The wings themselves contain mostly images labeled as non-aurora. Because there are only a handful of images labeled containing the moon, there are none of these in this randomly selected split present.

Images labeled as arcs show no preferred location in the mapping, whereas all other classes are grouped together in one or more clusters. The images labeled discrete for example, are mostly between the wings, the images labeled diffuse between the discrete aurora and the wings on each side and cloudy and clear images can be found in several groups of both wings.

The first selected area near (1, 11) in the coordinate system of the mapping contains most of the images labeled as arcs. The images belonging to this group show arcs like we have seen them in Figures 3a and 3b. The image in the top left is ambiguous and could also be labeled as clear, whereas the other images are correct. Area 3 to its bottom right near (3, 10) is close to these images in the feature space and also shows similar images but much of the visible aurora spans more of the image and more patches are visible in general. One of these images has been labeled incorrectly as cloudy, but also shows its features near the edge of the image. The other images have been labeled correctly as aurora but whether discrete or diffuse is the correct label for these images is an ambiguous choice.

The images to its top right near (5, 12) in Area 2 are similar to those in Areas 1 and 3, but have all been labeled as clear. The important difference seems to be that the brightness of the aurora in the crescent shape near the edges is low compared to the background, whereas the shapes in Area 1 have been bright against the background.

The images in Area 4 near (5, 8), Area 5 near (1, 8) and Area 11 near (−1, 6) have all been classified correctly as diffuse or discrete. They show high resemblance to the images classified with high confidence as presented in Figures 3b and 3c.

Area 7 near (−5, 6) shows images classified as clear with some diffuse images present in the area. Because the images have been normalised to their highest and lowest values and these values lie close together, they appear grainy. This means that the brightness of the visible aurora was very low. The classification as clear skies would align with a human classifier due to the low brightness if presented with the unprocessed image. In area 8 near (13.5, −1.5) we see an example of images taken of a clear night sky. This area is far away from any images labeled in auroral classes

The groups near areas 2, 7 and 8 give 3 clusters of images labeled mostly clear, that lie far apart but are classified correctly in their context. Those clusters show distinct differences in their images, which is represented in their features and therefore visible in the mapping. Only the images in area 8 are true images of a clear night sky, the other images are edge cases or near-edge cases that lie close to auroral images in this mapping. These images are also closest to the images classified with high confidence in Figure 3f.

While it is possible to draw boundaries around these clusters, especially in a higher dimensional space much like an SVM does, these cases are difficult to discern in the feature space of the images.

Table 6
Confusion Matrix in the Same Style as Table 3 With Manual Labels Used as Ground Truth for 1,330 Randomly Selected Images From Figure 6

	arc	diffuse	discrete	cloud	moon	clear
arc	3	3	19	3	0	3
diffuse	1	90	28	56	0	6
discrete	4	102	155	33	0	16
cloud	0	9	0	335	0	14
moon	0	0	0	0	0	0
clear	6	25	28	223	0	168

Note. Overall and two-class accuracy is 56% and 86%, respectively.

Area 6 near (11, 3) shows images labeled as cloudy. While the classification is correct, the images suggest that there are aurora visible behind the clouds.

Area 9 near (−5, 2) and area 10 near (−4, −3) also show images classified as cloudy. The images in area 9 suggest that there are again aurora behind the clouds, but this time the clouds seem to be thinner or the aurora brighter than in area 6. The clouds in area 10 show no visible aurora behind them.

At last area 12 near (−2, 3) is located between area 9 showing clouds and area 11 showing discrete aurora. Visually, the images placed here are a mixture between the two areas. The aurora are very diffuse or partially obscured by thin clouds which scatter the light and make them appear diffuse but also show those elongated bows with uneven brightness usually present in images classified with high confidence as discrete aurora. The misclassification of the image in the top right is most likely due to the big, fuzzy patch in the top right part of the image. The other images have been classified as diffuse, but it is difficult to determine only from the images whether they are truly diffuse or discrete and obscured by clouds.

Now that we have analyzed images in the clusters, we can see differences, that might be present in the features that are reflected in the mapping created by the UMAP algorithm. In the very middle between the wings (areas 5 and 11) bright structures on a dark background are mostly visible. Toward the right wing (ares 4, 6 and 8) these structures blend more and more with the background and the images become more homogeneous the further along the wing the images are located. On a microscale the images are homogeneous, moving toward macroscale, gradients are apparent.

Toward the left wing (ares 7, 12, 9 and 10), we also see blending with the background, but this blending is less homogeneous on a microscale and the grainy images indicate lower brightness of the images. Because the images are scaled to their highest and lowest brightness (see Section 3.1), only the relative brightness of features against the background can be extracted by the neural network.

We have shown that the UMAP algorithm already is able to reduce the 1,000-dimensional feature space of the images to a 2D image representation of the features where images are bundled in groups of visually similar images. We find that this often corresponds to correctly classified images, but that especially the grouping and classification of images in the edge case between clear skies and any aurora class are difficult to represent this way. If we were however to draw only a few straight lines through this image, a crude classification into the different classes could already be achieved.

Classification based on higher dimensional features should therefore yield better results. Still, edge cases have to be taken into account depending on the application. If for example, the goal was to remove all non-aurora images, based on the probabilities all images with combined probability for auroral images less than 90% could be discarded.

In Section 4.5 we will show two possible applications of the classifier, one of which could be used as a cloud-removal tool based on this principle.

4.4. Comparison to Human Classification

Out of the 3,000 images previously discussed, we have randomly selected 1,330 images for manual classification, which were given a single pass each and assigned one of the six classes. From this we obtain the confusion matrix as detailed in Table 6. Although there are no images detailing the moon in the set, we have kept the row and column for the corresponding classification for completeness. We see that many misclassified images show diffuse aurora and have been labeled as discrete aurora or vice versa. Images showing arcs are almost not present in the whole set. Misclassification between clouds and clear images is mostly only that way, that clear images have been predicted to contain clouds but not that cloudy images have been predicted as clear images. Similarly, images manually classified as containing some form of aurora have often been predicted as cloudy.

Table 7
Summary of Metrics for Table 6 in the Same Style as Table 4 for 1,330 Randomly Selected and Manually Classified Images From Figure 6

classes	precision	recall	f1-score	number of images in test set
arc	0.21	0.10	0.13	31
diffuse	0.39	0.50	0.44	181
discrete	0.67	0.50	0.57	310
cloud	0.52	0.94	0.66	358
moon	—	—	—	0
clear	0.81	0.37	0.51	450
Aurora	0.86	0.78	0.81	522
No Aurora	0.86	0.92	0.89	808

In the previous section we have already seen, that small parts of cloud coverage tend to lead the classifier to predict the images as cloudy. This is again confirmed in the high recall, but low precision value for cloud class as detailed in Table 7. This table gives some classification scores, obtained from the confusion matrix. The distribution of classes in the manually labeled images is similar to the distribution of classes obtained from the manual classifier, shown in Table 5. In our sample, 39% have been labeled as some form of aurora, the classifier attached an aurora class to 36% of images. For our representative sample size and with a significance level of $\alpha = 0.05$ this difference in classification is statistically significant. Performing the significance test we saw that a slightly smaller sample size, or just 10 more correctly classified images would have lead to an insignificant result.

A big difference is in the distribution of cloudy images compared to clear images in the non-aurora images. The classifier labeled 75% of these to be cloudy, in manual labeling only 44% were assigned this label.

For aurora classes, classifier and human labeling agree on the distribution of arcs (5% by classifier, 6% by human), but the classifier assigns the diffuse and discrete class equally often, while manual classification favors the discrete class at 59%–35% for diffuse aurora.

From the scores we can see that the classifier performed poorly compared to our manual classification in the six classes. Many of the images in the real set of data are not as clean as the images contained in the test set. Besides the ambiguity between the aurora classes and the "clear sky" class, which is a result of image features - mostly arcs - near the edge of the image, most images in the training set contain exactly one class of aurora. This class in itself might be ambiguous, which lead to most of the training error, but as we have shown in the previous section much of the real image data contains two classes, some might even contain three.

The two class scores perform similarly to the scores obtained for the testing set. The accuracy for two classes is at 86%, slightly lower than the 89% obtained on the test data, but there is a statistically significant difference in the distribution of the classes. This discrepancy could already be explained by a bias to rather exclude edge cases, because only 39 more images would need to be classified correctly to achieve the same accuracy.

In Figure 7 we show these manually labeled images in the same mapping as in Figure 6. The color scheme has been kept the same, only that this time the points are colored by the classes assigned by human classification. Those points marked by "." have been classified correctly in the two class classification, points classified incorrectly have been marked as squares. Comparing the clusters between both figures we see that the classes are on the same positions as before. The middle is dominated by images labeled "discrete", toward the outside "diffuse" images appear and the wings are made up of cloudy and clear images. Images labeled as "arcs" still do not follow a pattern or group together. Whereas the extended parts of both wings have been clearly classified cloudy before, now many images are labeled as clear as well.

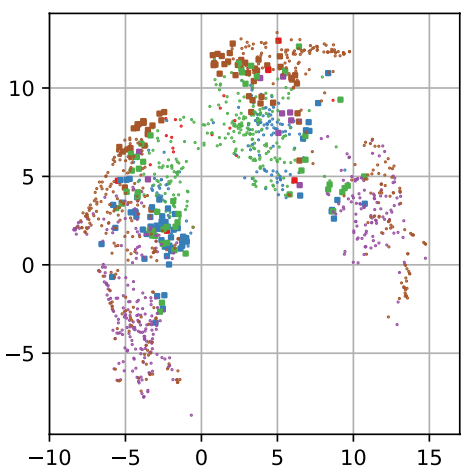

Figure 7. Manually attached labels for 1,330 randomly selected images from 6. The color scheme and coordinate system is the same as in the previous figure. Points marked by "." have been classified correctly in the two class classification, points classified incorrectly have been marked as squares.

The wings still contain mostly images which have been classified correctly in the two class case, the same is true for the middle of the image. The crescent shaped boundaries between these regions contain 260 of the 285 images, which have been classified incorrectly. All remaining 25 images have been manually labeled as some form of aurora and have gone undetected by the classifier.

It is these edge cases, which account for the low accuracy of the classification. Although there is a statistical difference between the classifications, there are no apparent systematic problems in the classification that would be evident in Figure 7 The classification errors are therefore similarly to differences arising if the same images would be given to two human classifiers, each with their own bias for edge cases. The same way humans ambiguously

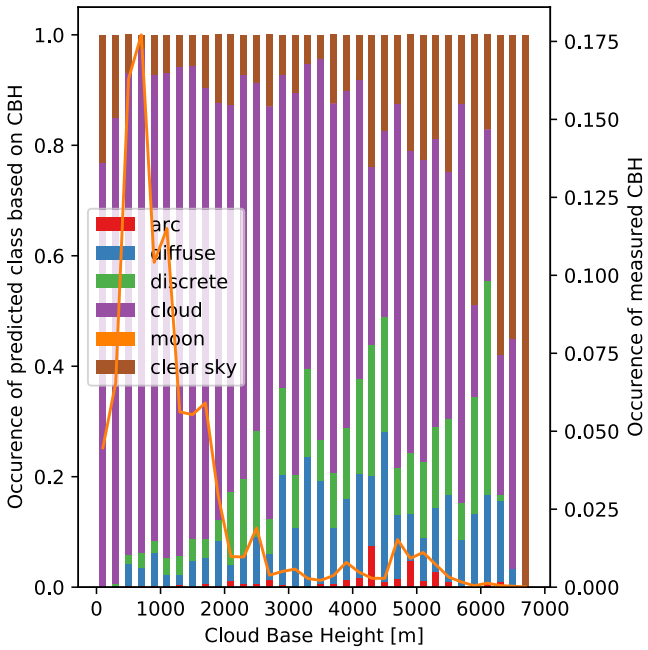

Figure 8. Distribution of predicted classes by measured Cloud Base Height (bar graphs) and occurrence of different measured heights (orange line).

differentiate between aurora and non-aurora the classifier has been trained, makes decisions and mistakes.

Because the classifier also returns the probabilities of each class for each image, it can be finetuned for high precision (e.g., "reject every aurora image below 90% probability") or high recall (e.g., "accept every aurora image above 10% probability"). It could even be quickly retrained by relabeling the OATH training images according to one's needs. We will show below how finetuning can be applied by comparing our predictions for the "cloud" class with meteorological data.

Overall we see good agreement between the classifier and manual classification when distinguishing between images showing aurora and images not showing aurora. In the six-class case we see more problems when distinguishing between different aurora classes and the classifier's tendency to label images "cloudy" even if only small cloud coverage is present. We attribute this to the fact that many images show features of different classes, for example, diffuse and discrete aurora, covered in some clouds all in one image.

4.5. Potential Applications

To show applications beyond image classification we have selected two tasks. The first will be comparing our prediction of whether an image is cloudy to available meteorological data, the second will be modeling the perturbation in Earth's magnetic field as measured locally for every image.

Meteorological data in the form of the Cloud Base Height (i.e., the lowest part of a cloud directly above the ceilometer) is unique for us in such a way that we can obtain a ground truth for whether it is cloudy or not. For the other predictions there is no such way to obtain a reliable comparison. The reliability of our classifier with regards to the cloud prediction is not necessarily related to the reliability of the prediction for auroral classes, but good performance in this task suggests that prediction of other classes may perform equally well. Additionally, if this performs well, it can be applied to reliably filter and remove cloudy images from unclassified datasets before further processing.

Magnetometer data is not related directly to any measurable quantity in the images, but as explained in the introduction, northern lights and variations in the Earth's magnetic field show a strong physical connection. The aurora are the optical manifestation of how the Earth's magnetosphere is influenced by the solar wind, and how ionospheric currents due to energetic particles influence the measurements of the Earth's magnetic field. Being able to connect images to physical measurements could allow researchers to analyze many images at once or enable large scale statistical analysis of images. Studies that need to analyze many images could therefore be sped up or expanded to larger timescales or missing data could be extrapolated from auroral images.

Based on the features extracted by the neural network we therefore try to predict values for the perturbations in the Earth's local magnetic field as measured by the magnetometer collocated to the all sky imager.

4.5.1. Predicting Cloud Coverage

We are using the classifier that we previously trained on the OATH images without retraining it on the ceilometer data. The images and ceilometer data of this section belong to a new set of data unknown to the classifier at the time of training.

It is important to note that the ceilometer only measures the Cloud Base Height (CBH) directly above it, therefore the prediction and measurement might disagree slightly, but it is still good for a general evaluation. In Figure 8 we see the distribution for the predicted classes based on the measured cloud base height. Only 195,620 of the total 550,286 were assigned a measured cloud base height, for the remaining images the measurement returned "cloud free". Of those, 165,765 lie below a measured cloud base height of 2,000 m. Above this height, the amount of clouds measured for these heights reduces a lot and the classifier more often predicts northern lights or a clear

Table 8
Prediction of Cloudy by the Classifier Compared to the Values Obtained by the Ceilometer

Classes	Precision	Recall	F1-score	Number of images
Not cloudy (CBH > 2,000 m)	0.92	0.68	0.78	384,521
Cloudy (CBH ≤ 2,000 m)	0.54	0.86	0.66	165,765

Note. The total accuracy is 74%.

sky. We have therefore chosen a cut-off of 2,000 m, below which we consider an image to be "cloudy" and above which it is not cloudy.

In Table 8 we see the accuracies of the classifier when comparing to the thresholded cloud base height classifier. This way only 30% of the images are classified as cloudy, whereas our classifier predicted almost 50%. As we have shown in the previous section, the classifier assigns the cloudy label already to images with small patches of cloud but is otherwise good at assessing cloudiness. The classifier overestimating the cloudiness is therefore probably not the only factor for this discrepancy, but rather that the point-measurement of the ceilometer is not as good an estimator of cloud average as presumed.

Overall can see that this way non-cloudy images are selected with a precision of up to 92%, albeit with a recall of only 68%. This suggests that, while the ceilometer and the classifier disagree for patchy clouds, they agree on homogeneous cloud coverage. Because the classifier not only returns the final assigned class, but also the probabilities for each class, we can use the probability for each image to be "cloudy". Ignoring all other probabilities, we can slide the threshold above which an image is classified as "cloudy" by our classifier. For each threshold we calculate the accuracy of the prediction, that is, how often the predicted label agrees with the label assigned by the CBH extracted from the ceilometer.

The result of this is shown in blue in Figure 9a. For a low threshold, every image is classified as cloudy and we achieve an accuracy of about 30%, equal to the amount of cloudy images. The maximum accuracy of 85.6% is achieved at a threshold of 75.5%. This is a significant improvement of the accuracy of 73.6% achieved without adaptive thresholding. Increasing the threshold further, the limit goes toward classifying everything as non-cloudy, which is the case in about 70% of the images. If we were to randomly predict an image to be either cloudy or not at a probability of 30:70 according to the classes' distribution in the dataset, we would obtain a curve as shown in orange in Figure 9a. The shape of the curve based on our thresholded prediction suggests that our classifier performs significantly better than randomly guessing and that the two classes are well separated by a threshold near 50% although there are twice as many non-cloudy images as there are cloudy images.

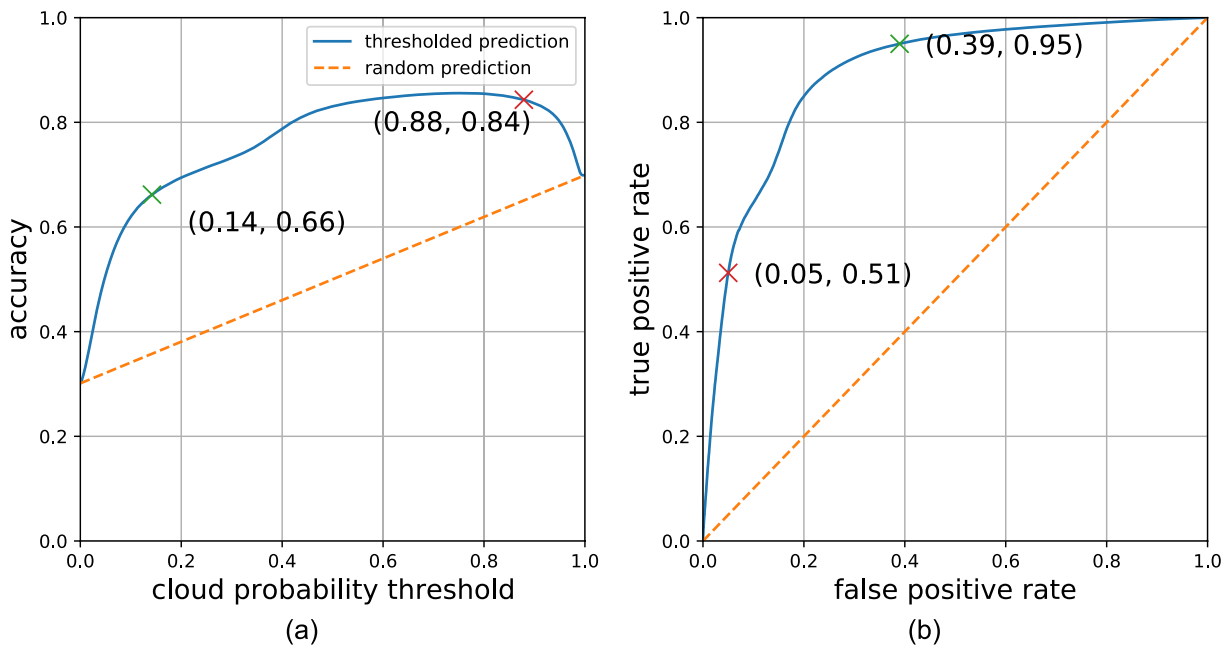

Figure 9. Assessment of the prediction quality for "cloudy" vs. "non-cloudy". Panel (a) shows the accuracy as a function of the threshold and Panel (b) shows the ROC curve. In both plots, the continuous blue line shows our thresholded prediction and the orange dotted line corresponds to a random classifier. In the thresholded prediction of Panel (a), a small threshold of 0 is equivalent to predict every image as being cloudy (giving 30% accuracy which is equal to the overall fraction of cloudy images). A large threshold of 1 is equivalent to predict every image as being non-cloud (giving 70% accuracy which is equal to the overall fraction of non-cloudy images). The two distinguished working points of both Panels correspond to a "cloud removal" (green) and "cloud extraction" (red) application scenario.

In Figure 9b we show the true positive rate plotted against the false positive rate. Because we use the ceilometer as ground truth, an image is a true positive, if both ceilometer and classifier label it "cloudy" and false positive, if the ceilometer labels "not cloudy" but the classifier labels "cloudy". We see a typical receiver operating characteristic (ROC) curve with an area under the curve of 0.89. The ROC curve summarizes how well the two classes' distributions can be separated by the classifier and can be used to tune the classifier to the desired thresholds.

On both curves we have marked two possible scenarios for how to tune the classifier when using it to filter out cloudy images in a dataset.

For the red mark, the false positive rate was set to 5%. This corresponds to setting the threshold to 0.88 for which a total accuracy of 84% was achieved. In this case, 51% of the cloudy images would be identified correctly and the remaining 51% of cloudy images would go undetected. This could be a desirable setting to remove the most cloudy images, if some filtering is required but the following processing is robust enough to handle some cloudy images or if the sample size of the remaining images should be kept as large as possible.

The green marking shows the opposite case. Here, the true positive rate was set to 95%, meaning that almost every cloudy image was detected. Here the threshold is set to 0.14 and returns a total accuracy of 66%. This however leads to falsely labeling 39% of the non-cloudy images as well. This approach could be used when rigid filtering is required at the cost of reducing the remaining sample size.

Overall we see that our classifier compares well to real-world data where we have a ground truth available to compare it to. It will be useful for filtering if there are no other means of filtering available or to support other filtering methods. The accuracy of this method will however depend on the distribution of classes in the dataset it is applied to. Depending on the application, an initial threshold for filtering can be chosen based on our ROC curve but should be refined on some manually labeled test cases. Retraining the classifier on a pure "cloudy" vs. "not cloudy" basis could also further improve this technique.

4.5.2. Predicting Magnetometer Values

To show that the features can be used beyond image classification we try to predict perturbations in the Earth's local magnetic field. For this, each image is assigned the value measured by the magnetometer closest in time. With a resolution of 60 s, the difference in time between when an image was taken and its assigned magnetometer measurement is between 0 to 30 s. To define perturbations, we assume a constant baseline value and therefore remove the mean. Although the Earth's magnetic field and therefore the baseline changes over the years, we assume it is near constant for our short timespan of 4 months. We only used images and their respective features where our classifier placed the images into one of the categories showing aurora and where the Cloud Base Height was above 2,000 m. This gives 188,458 images for analysis out of originally 550,286 within the timeframe November 2010–February 2011. This is even more restrictive than relying solely on the classifier's prediction to ensure our dataset contains as little noise as possible at the cost of removing some otherwise good images.

We again split training and testing data consecutively, not randomly, to minimize possible bleeding of information from the training data into the testing data. We show the resulting distribution of training and testing data in Figure 10a. For values above −400 nT the training and testing dataset are close to the distribution of the original data. Values below are only present in the training data. This is because there was only one event of this magnitude in the given timeframe. This part of the distribution can therefore only be present in either training or testing data, not in both sets at the same time. We then trained a linear Ridge Model on the data. In Figure 10b we show the prediction of the magnetic field perturbations for an event on 02-05-2011 between 18:30 and 22:00 UTC corresponding to 1,558 of 55,409 values in the testing dataset. In the whole set of testing data, this was the only event which reached perturbations below −50 nT for a significant period of time. Other events were either weaker or only present in few consecutive images.

The blue dots correspond to the testing values, the orange dots are the values predicted by our linear ridge model. The green line is a Savitzky-Golay filter of order 5 and window length 51 through our predictions and the red line the absolute error between the Savitzky-Golay filtered data and the testing data. The event shows three distinct negative peaks of the disturbance of down to −400 nT, each at around 18:50, 20:00 and 21:30 UTC. We can see that individual predicted values can match their respective testing values. Especially the second and the beginning of the third event (19:45-21:00 UTC) can be matched well by individual predictions. The predicted data however shows high variance.

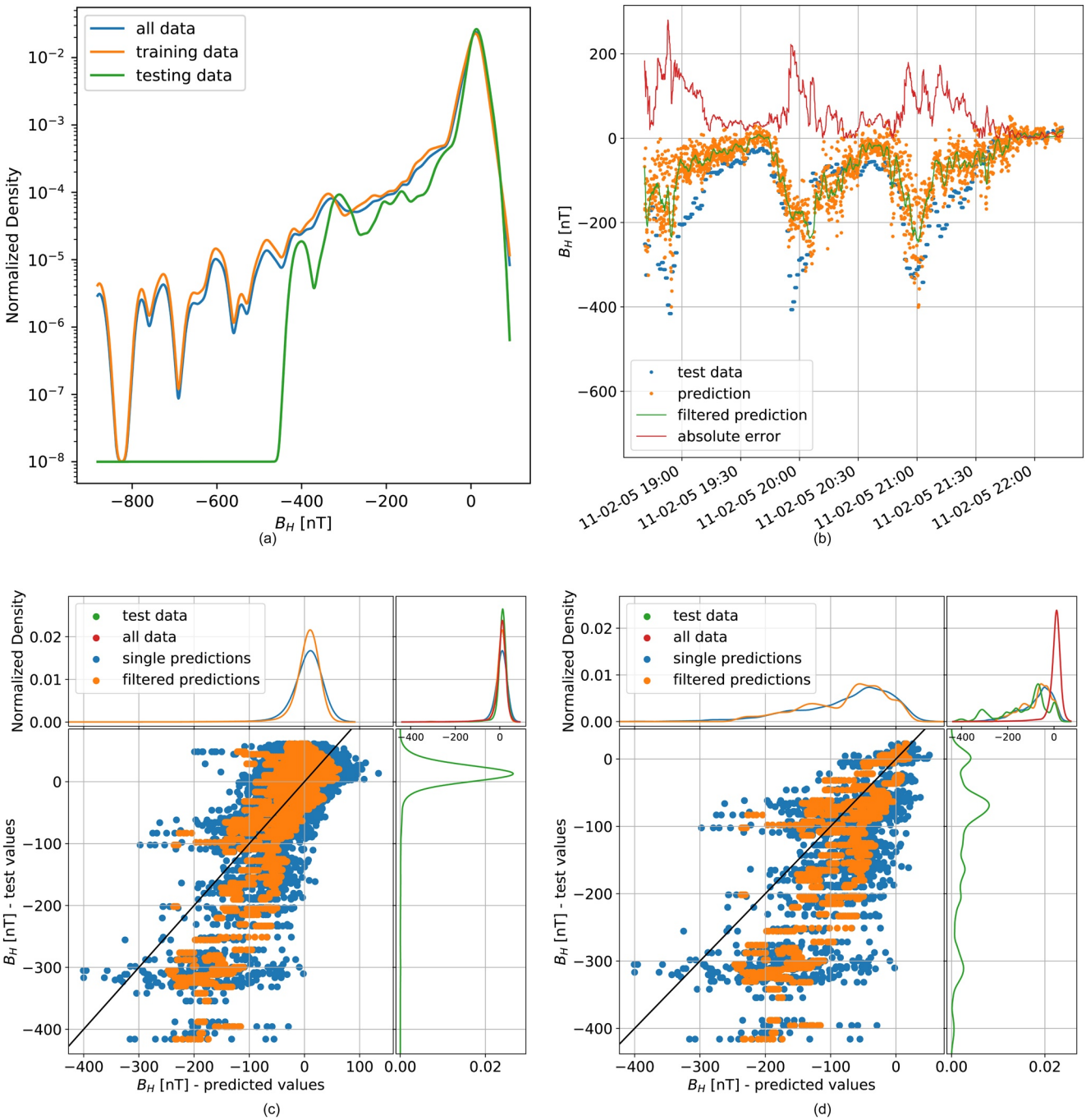

Figure 10. Predictions of the magnetic field disturbance in the horizontal direction for the 2010/ 2011 season, based on features extracted by the neural network. Panel (a) shows the normalised densities of our training and testing splits compared to the densities of the whole set of data. Panel (b) shows a selected event on 02-05-2011 between 18:30 and 22:00 UTC. Panel (c) shows a scatter plot for single predictions and smoothed values against their respective testing values, for all testing values. The graphs above and to the right show the normalised densities of the distributions, the graph in the top right a combination of all previous distributions and the whole set of data. Panel (d) shows a figure in the same fashion as (c), this time only for the selected event from (b) instead of for the whole set of testing data.

To remove some of this variance, we applied a Savitzky-Golay filter of order 5 and window length 51. To match the peaks better, a shorter window length would be required and to filter out the variance of the near-constant data better, a longer window length would be required This makes this a compromise for evaluation, not a perfect tool for the prediction.

Table 9
Mean Squared Errors and Absolute Errors for Single Values and the Running Average of the Prediction for the Whole Season and 3 Selected Events

				Mean squared error (nT^2)		Mean absolute error (nT)		Coefficient of determination (R^2)	
	Figure	Event length (# of test-datapoints)	Standard deviation of test data (nT)	Single value	Savitzky-Golay filter	Single value	Savitzky-Golay filter	Single value	Savitzky-Golay filter
All testing data		56,493	35.8	920.2	626.0	21.1	16.3	0.28	0.51
Selected testing data	10b	2,030	105.2	8319.6	7045.7	68.7	64.9	0.25	0.36

We can see that in general our predictor underestimates the magnitude of the perturbations but manages to follow the direction. Between 18:30 and 19:30 UTC the filtered data follows the testing data into the peak and matches the point in time where the test data reaches its minimum. It is able to detect that the aurora following afterward at 19:15-19:40 UTC correspond to no meaningfull magnetic disturbance, before it follows the data into the next minimum at 20:00 UTC. This time the model does not underestimate the magnitude of the event but does not match it as well in time as before. This is again followed by a correctly identified calm event where between 20:00 and 20:30 UTC the absolute error of the filtered data reaches near zero, while the magnitude of the data is still around −100 nT.

The last event at 21:00 UTC can again be matched in time, while the magnitude is underestimated. The timing of the secondary peak at 21:15 UTC is also matched well, but underestimated in magnitude. Afterward the predicted values drop off together with the testing data to zero.

In Figures 10c and 10d we show how the distributions of predicted values behave against their corresponding true values. Figure 10c shows this for the whole range of testing data, Figure 10d only for the selected event we described before. The scatterplot in the bottom left shows the predicted disturbance against their true values, once for the single predictions (blue) and once for the filtered predictions (orange). The black line is drawn where $B_{\text{test}} = B_{\text{prediction}}$. On top the normalised density of the single and filtered predictions calculated by a kernel density estimate with Gaussian kernel and bandwidth of 10 is shown, to the right is the same plot for the testing data. In the top right corner the three densities are overlaid, together with the density of the whole dataset.

For the whole set (Figure 10c) most of the data is near the origin. We see that the single and filtered distribution of the testing data are nearly identical. The filtered distribution is narrower and shows more data near its mean value, because the filter removes outliers. The filtered data however still seems to represent the originally predicted data well enough. Overlaid with the testing data and data of the whole dataset we see that all distributions are nearly identical.

The plot in Figure 10d belongs to the previously shown event. The distribution of testing data shows an almost even profile over the whole possible range of values. The single and filtered predictions are again close enough that the filtered prediction is an accurate representation of the single values with less variance. The distributions show more data toward lower negative values and their values of maximum intensity is below that of the full distribution. This was to be expected, because we chose this set specifically because it showed large perturbations.

Comparing the predicted distributions to the testing distribution we see that the distribution shows more values toward predictions of lower magnitude, predictions of high magnitude are underrepresented in the predicted distribution. This is also reflected in the scatter plot. Ideally every point would fall on the black line. For points with $B_{\text{test}} \geq -100$ nT, the filtered predictions are closer to the line, whereas the single predictions show larger variance. Toward larger negative test values, the predicted values underestimate the testing values in most cases. This suggests that the predictor can predict the direction of the disturbances, but because most of the training data is based on low disturbances, it is biased toward predicting events with lower magnitude. This can be improved by providing the predictor with more training data. However as we explained before, this has to be done carefully, to find a good representation of the baseline over several years from which the perturbations can be defined. Measurements from different cameras and stations can also be used, but again the values have to be prepared carefully.

In Table 9 we give the errors and coefficients of determination obtained for the selected event and the whole season. For the whole testing set as well as the selected event, the predictor performed better when comparing the filtered data to the true values. For all testing data, the mean squared error is below the squared standard deviation ($35.8^2 = 1281.6$)

as well as for the selected event ($105.2^2 = 11067.0$). For the mean absolute error we see similarly that the errors are below the standard deviation and that the filtered values perform better. For the whole set we reach a coefficient of determination of up to $R^2 = 0.51$ for the filtered data. Overall, we find that that our model cannot predict the perturbations on a single-image-basis but when removing variance from the predicted data we find that the filtered values follow the direction of the perturbations but in general underestimate them. For small perturbations below a magnitude of 100 nT the distribution of predicted data matches that of the testing values, because 98% of the values lie within this range and the predictor therefore has much training information available. To more accurately model the larger perturbations, the time series nature of the data could be exploited, but more data to train the predictor on is needed as well.

We can also conclude that information beyond that necessary for image classification is present in the extracted features, although we used a neural network which was specifically trained for classification. Because all sky imagers are often paired with magnetometers, no manual processing (like labeling images) would be required to create a dataset consisting of hundreds of millions image-magnetometer-measurement pairs.

While we used transfer learning methods, because our original set only contained 5,824 images, a dataset of that size could be used to train or finetune neural networks specific to tasks like these. This could in turn be once more used to assist image analysis by searching for specific, interesting events, fill gaps or interpolate in data or together with other sources be used for space weather prediction.

4.6. Future Community Efforts

Although the improvement in classification of all-sky images seems minor, we believe there is much potential in this method.

With only 5,824 images available for training and testing we achieve precision, recall and accuracy scores around 90% when distinguishing between aurora and non-aurora, making the classifier suitable for removing non-aurora images from large datasets. For classification between auroral classes, there is still an improvement possible, but our extensive validation efforts minimize the probability that we overestimate the accuracy of our classifier.

Because our classifier is based on features extracted by a pretrained and unmodified neural network, these features are universal and can be stored space-efficiently. The computationally expensive part of feature-extraction therefore has to be done only once. On our machine we manage to extract features of approximately 3 million images per day. Classifying these images based on their extracted features works at a rate of up to 100 million images per day, which can be improved by parallelizing the task. If a better classifier becomes available, the stored features can be recalled and images re-classified quickly.

We argue that with only minor effort from the community, enough labeled image data can be produced to create a universal all-sky image classifier, which can be redistributed and applied to the extracted features.

5. Conclusion and Outlook

Based on the Oslo Auroral THEMIS (OATH) dataset, containing 5,824 labeled all-sky images in six categories - “arc,” “diffuse,” “discrete,” “cloud,” “moon,” “clear sky/no aurora,” we have evaluated the performance of 80 pretrained neural networks in terms of speed of feature extraction and performance of the extracted features for classification using a ridge classifier. The best performing network's features have been used to train a linear support vector machine, where the best hyperparameters have been evaluated using tenfold cross validation. We achieve $73.00 \pm 5.22\%$ accuracy in the six-class classification, and $91.27 \pm 2.83\%$ when aggregating images into two classes “aurora” and “no aurora” after classification.

While these values are significantly lower than previous publications, we carefully validated our results, to ensure our reported metrics will hold up against unseen data. These steps were not undertaken in previous publications. When applying our method to select training and testing data to a recent and similar study, the accuracy decreases to a level below our findings. This shows the importance of our validation efforts. For our test images, the biggest errors arise from misclassification between auroral classes among each other and between the auroral classes and “clear sky”. Both problems can be shown to originate in the training data, where images cannot be identified to clearly belong only in either class, resulting in ambiguity in the labels.

We show the application of our classifier to 550,000 previously unlabeled images taken in Ny-Ålesund between Nov 2010 and Feb 2011. To test the classifier, we analyze its confidence in several subsets of the data. We show that images are assigned their correct labels with high confidence but that there are also images where aurora is only visible in small parts of the sky that are labeled ambiguously by the classifier, most likely because of the same problem in the training data as for the testing data. A comparison to manually labeled images also reveals good agreement for unambiguous cases but also confirms differences for ambiguous cases.

Expanding the training data by increasing the size of the dataset with labeled images from different sources and allowing multi labeled images (e.g., "arcs & clouds") or using a multi-stage classifier could lead to an improvement in prediction. Because the transfer learning approach works on small amounts of training images already, the classifier can also easily be modified based on the application. One could for example, introduce more training images specifically designed to detect another class of aurora or phenomenon and repurpose the features extracted previously.

It might also be interesting to consult in-situ particle observations or imagery data from which particle information can be inferred, such as the Defense Meteorological Satellite Program satellite SSJ and SSUSI instrument observations and compare these to the classification of images. This could yield further insights about which images and aurora types are classified correctly under which conditions.

To better understand the extracted features, we employ UMAP to create a 2D representation of the 1000-dimensional feature space. This showed that optically similar images lie close together in the mapped feature space.

To show physical application beyond classification and test our classifier's performance on real-world data, we compared the predictions of the "cloudy" class to meteorological data and predict perturbations in the Earth's local magnetic field based on the extracted features.

We find that the classifier accurately distinguishes between cloudy and non-cloudy images in 82% of the images, performing better than randomly guessing. The output of the classifier can easily be finetuned to be used as a filter to remove cloudy images from a dataset with desired sensitivity or specificity.

When predicting magnetic disturbance, we find that the predictive capability of a single image is not enough to accurately predict the magnetic field values, however a Savitzky-Golay Filter is able to accurately follow the measured values' trend. This shows the rudimentary connection between the features extracted from the images and underlying physical phenomena. Because auroral images are taken as a time series, modeling of magnetic field values could be improved by employing a model that takes the time-series nature of the data into account.

We have therefore shown that the approach demonstrated in a previous paper is viable for large scale application. All our methods use publicly available, tested and trustworthy python libraries (PyTorch and scikit-learn), which are already used in many other scientific environments and can be applied in just a few lines of code. This use of standardised, open-source software allows for a collaborative effort in the community, for example, to create standardised classification methods. Extracted features are stored space efficiently and can be transferred more easily. Once extracted, they can also be stored and retrieved much faster than images.

The images used for training and the images the classifier was applied to are from different origins, are of different size and are in a different format but were compatible to each other without requiring major modifications to the code. In the spirit of open science and to foster collaboration, we make our classifier publicly available under the Attribution 4.0 International (CC BY 4.0) license. This will allow anyone to access the feature extractor and apply our classifier and could facilitate global cooperation toward a common set of classifiers.

Appendix A: Overview of Tested Neural Networks

Table A1 shows all tested pretrained neural networks. The data is taken from the "times.csv" file we provide in machine-readable format alongside. This data can therefore be used to recreate our extraction techniques without manual modifications to the contents of the file. The "model" column gives the name of the model as used by torchvision or pretrained-models.pytorch. and the origin of the model. The suffix "_cadene" marks models from pretraine-models.pytorch (Cadene, 2020) and "_torchvision" those from PyTorch's (Paszke et al., 2019) torchvision (https://pytorch.org/vision/stable/index.html). The "key" column gives the name of the dataset this network was trained on. In most cases, this is the "imagenet" dataset, but some networks may differ. "num_classes" is

Journal of Geophysical Research: Space Physics
10.1029/2021JA029683

Table A1
Overview of all Tested Pretrained Neural Networks

	Model	Key	num_classes	Size	Mean	Std	Diff	Mem	acc_test_class2	dev_test_class2	acc_train_class2	dev_train_class2	acc_test_class6	dev_test_class6	acc_train_class6	dev_train_class6
0	fbresnet152_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	68.2068	2175.0000	0.9250	0.0127	0.9618	0.0038	0.7099	0.0435	0.8542	0.0139
1	bninception_cadene	imagenet	1,000	224	[104. 117. 128.]	[1. 1. 1.]	45.9797	3369.0000	0.8540	0.0430	0.8967	0.0099	0.6219	0.0642	0.7290	0.0182
2	resnext101_32x4d_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	65.0266	3369.0000	0.9250	0.0196	0.9668	0.0044	0.7149	0.0517	0.8675	0.0145
3	resnext101_64x4d_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	83.1215	4153.0000	0.9196	0.0138	0.9613	0.0042	0.7013	0.0539	0.8671	0.0133
4	inceptionv4_cadene	imagenet	1,000	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	73.1479	4153.0000	0.9044	0.0276	0.9641	0.0066	0.6716	0.0496	0.8598	0.0120
5	inceptionv4_cadene	imagenet + background	1,001	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	71.0594	4153.0000	0.9045	0.0276	0.9641	0.0066	0.6716	0.0496	0.8599	0.0120
6	inceptionresnetv2_cadene	imagenet	1,000	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	77.2813	4189.0000	0.9195	0.0175	0.9675	0.0045	0.7082	0.0460	0.8729	0.0106
7	inceptionresnetv2_cadene	imagenet + background	1,001	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	78.1985	4189.0000	0.9194	0.0175	0.9675	0.0045	0.7082	0.0461	0.8729	0.0106
8	alexnet_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	40.4227	4189.0000	0.8980	0.0281	0.9552	0.0070	0.6891	0.0543	0.8523	0.0153
9	densenet121_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	48.8951	4189.0000	0.9275	0.0195	0.9673	0.0064	0.7136	0.0474	0.8726	0.0147
10	densenet169_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	51.6538	4193.0000	0.9260	0.0166	0.9678	0.0048	0.7148	0.0563	0.8804	0.0131
11	densenet201_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.1744	4257.0000	0.9262	0.0171	0.9708	0.0054	0.7037	0.0501	0.8803	0.0124
12	densenet161_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	64.7054	4313.0000	0.9325	0.0160	0.9721	0.0046	0.7127	0.0481	0.8891	0.0127
13	resnet18_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	43.8311	4313.0000	0.9254	0.0207	0.9611	0.0056	0.7042	0.0443	0.8454	0.0126
14	resnet34_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	46.2278	4313.0000	0.9137	0.0187	0.9608	0.0045	0.6977	0.0491	0.8481	0.0150

Journal of Geophysical Research: Space Physics

10.1029/2021JA029683

Table A1
Continued

	Model	Key	num_classes	Size	Mean	Std	Diff	Mem	acc_test_class2	dev_test_class2	acc_train_class2	dev_train_class2	acc_test_class6	dev_test_class6	acc_train_class6	dev_train_class6
15	resnet50_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	50.5016	4321.0000	0.9199	0.0234	0.9633	0.0052	0.7029	0.0592	0.8599	0.0110
16	resnet101_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	59.4153	4373.0000	0.9289	0.0140	0.9635	0.0042	0.7120	0.0464	0.8587	0.0128
17	resnet152_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	63.5194	4455.0000	0.9327	0.0188	0.9668	0.0050	0.7151	0.0499	0.8604	0.0132
18	inceptionv3_cadene	imagenet	1,000	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	59.1663	4493.0000	0.9127	0.0205	0.9681	0.0050	0.6995	0.0437	0.8870	0.0127
19	squeezenet1_0_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	44.2880	4495.0000	0.9186	0.0186	0.9701	0.0056	0.7185	0.0518	0.8879	0.0118
20	squeezenet1_1_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	42.5836	4497.0000	0.9087	0.0288	0.9691	0.0053	0.7139	0.0570	0.8896	0.0118
21	vgg11_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	51.5564	5281.0000	0.9161	0.0282	0.9590	0.0070	0.6904	0.0551	0.8503	0.0129
22	vgg11_bn_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	49.1940	6065.0000	0.9140	0.0216	0.9552	0.0068	0.7005	0.0500	0.8423	0.0194
23	vgg13_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	52.7669	6065.0000	0.9088	0.0304	0.9584	0.0056	0.7009	0.0540	0.8530	0.0148
24	vgg13_bn_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	54.1080	6849.0000	0.9186	0.0250	0.9566	0.0056	0.6934	0.0498	0.8388	0.0174
25	vgg16_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	56.4526	6849.0000	0.9017	0.0231	0.9486	0.0067	0.6823	0.0456	0.8397	0.0149
26	vgg16_bn_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.4583	6849.0000	0.9242	0.0177	0.9545	0.0047	0.6934	0.0464	0.8264	0.0156
27	vgg19_bn_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	58.5755	6849.0000	0.9187	0.0231	0.9517	0.0075	0.6954	0.0472	0.8227	0.0129

Journal of Geophysical Research: Space Physics

10.1029/2021JA029683

Table A1
Continued

	Model	Key	num_classes	Size	Mean	Std	Diff	Mem	acc_test_class2	dev_test_class2	acc_train_class2	dev_train_class2	acc_test_class6	dev_test_class6	acc_train_class6	dev_train_class6
28	vgg19_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	56.6498	7633.0000	0.9073	0.0299	0.9512	0.0068	0.6853	0.0506	0.8283	0.0156
29	nasnetamobile_cadene	imagenet	1,000	224	[0.5 0.5 0.5]	[0.5 0.5 0.5]	50.6985	7633.0000	0.9171	0.0275	0.9617	0.0058	0.6961	0.0527	0.8607	0.0108
30	nasnetalarge_cadene	imagenet	1,000	331	[0.5 0.5 0.5]	[0.5 0.5 0.5]	159.2706	11131.0000	0.9144	0.0253	0.9685	0.0046	0.7032	0.0540	0.8783	0.0126
31	nasnetalarge_cadene	imagenet + background	1,001	331	[0.5 0.5 0.5]	[0.5 0.5 0.5]	161.6795	11131.0000	0.9144	0.0252	0.9684	0.0046	0.7029	0.0537	0.8784	0.0125
32	dpn68_cadene	imagenet	1,000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	52.6899	11131.0000	0.9158	0.0210	0.9594	0.0070	0.7184	0.0542	0.8468	0.0156
33	dpn68b_cadene	imagenet+5k	1,000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	52.3839	11131.0000	0.9222	0.0154	0.9638	0.0058	0.7283	0.0481	0.8601	0.0137
34	dpn92_cadene	imagenet+5k	1,000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	62.0724	11131.0000	0.9288	0.0168	0.9678	0.0037	0.7094	0.0526	0.8511	0.0158
35	dpn98_cadene	imagenet	1,000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	77.0900	11131.0000	0.9133	0.0183	0.9604	0.0062	0.6993	0.0530	0.8409	0.0143
36	dpn131_cadene	imagenet	1,000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	88.3290	11131.0000	0.9085	0.0233	0.9609	0.0066	0.6910	0.0550	0.8423	0.0166
37	dpn107_cadene	imagenet+5k	1,000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	93.2658	11131.0000	0.9188	0.0181	0.9613	0.0061	0.6870	0.0503	0.8413	0.0131
38	xception_cadene	imagenet	1,000	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	71.9872	11131.0000	0.9205	0.0171	0.9684	0.0052	0.7048	0.0487	0.8807	0.0121
39	senet154_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	101.3283	4105.0000	0.9086	0.0212	0.9750	0.0041	0.6986	0.0532	0.8946	0.0094
40	se_resnet50_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	50.6819	4105.0000	0.9251	0.0171	0.9760	0.0044	0.7030	0.0478	0.9077	0.0095
41	se_resnet101_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	57.3070	4123.0000	0.9331	0.0156	0.9754	0.0041	0.7068	0.0456	0.8979	0.0086
42	se_resnet152_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	67.7151	4159.0000	0.9200	0.0157	0.9729	0.0033	0.7030	0.0514	0.8971	0.0096

Journal of Geophysical Research: Space Physics
10.1029/2021JA029683

Table A1
Continued

	Model	Key	num_classes	Size	Mean	Std	Diff	Mem	acc_test_class2	dev_test_class2	acc_train_class2	dev_train_class2	acc_test_class6	dev_test_class6	acc_train_class6	dev_train_class6
43	se_resnext50_32x4d_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.1590	4159.0000	0.9172	0.0135	0.9752	0.0032	0.6948	0.0461	0.9006	0.0082
44	se_resnext101_32x4d_cadene	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	66.5890	4159.0000	0.9265	0.0192	0.9767	0.0049	0.7065	0.0559	0.8897	0.0105
45	cafferesnet101_cadene	imagenet	1,000	224	[102.9801 115.9465 122.7717]	[1. 1. 1.]	55.7170	4159.0000	0.8454	0.0418	0.8774	0.0106	0.6002	0.0531	0.6971	0.0138
46	pnasnet5large_cadene	imagenet	1,000	331	[0.5 0.5 0.5]	[0.5 0.5 0.5]	158.9650	8331.0000	0.9002	0.0158	0.9615	0.0047	0.6733	0.0380	0.8504	0.0127
47	pnasnet5large_cadene	imagenet + background	1,001	331	[0.5 0.5 0.5]	[0.5 0.5 0.5]	158.2496	8331.0000	0.9001	0.0157	0.9614	0.0048	0.6733	0.0382	0.8504	0.0128
48	polynet_cadene	imagenet	1,000	331	[0.485 0.456 0.406]	[0.229 0.224 0.225]	136.7295	9367.0000	0.9340	0.0131	0.9680	0.0045	0.7170	0.0477	0.8708	0.0120
49	alexnet_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	40.2826	9367.0000	0.8980	0.0281	0.9552	0.0070	0.6891	0.0543	0.8523	0.0153
50	vgg11_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	51.1166	9367.0000	0.9161	0.0282	0.9590	0.0070	0.6904	0.0551	0.8503	0.0129
51	vgg11_bn_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	48.9552	9367.0000	0.9140	0.0216	0.9552	0.0068	0.7005	0.0500	0.8423	0.0194
52	vgg13_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	52.3303	9367.0000	0.9088	0.0304	0.9584	0.0056	0.7009	0.0540	0.8530	0.0148
53	vgg13_bn_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	54.1776	9367.0000	0.9186	0.0250	0.9566	0.0056	0.6934	0.0498	0.8388	0.0174
54	vgg16_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.1364	9367.0000	0.9017	0.0231	0.9486	0.0067	0.6823	0.0456	0.8397	0.0149
55	vgg16_bn_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	59.0248	9367.0000	0.9242	0.0177	0.9545	0.0047	0.6934	0.0464	0.8264	0.0156
56	vgg19_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	56.4227	9367.0000	0.9073	0.0299	0.9512	0.0068	0.6853	0.0506	0.8283	0.0156

Journal of Geophysical Research: Space Physics
10.1029/2021JA029683

Table A1
Continued

	Model	Key	num_classes	Size	Mean	Std	Diff	Mem	acc_test_class2	dev_test_class2	acc_train_class2	dev_train_class2	acc_test_class6	dev_test_class6	acc_train_class6	dev_train_class6
57	vgg19_bn_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	57.9345	9367.0000	0.9187	0.0231	0.9517	0.0075	0.6954	0.0472	0.8227	0.0129
58	resnet18_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	42.8714	9367.0000	0.9254	0.0207	0.9611	0.0056	0.7042	0.0443	0.8454	0.0126
59	resnet34_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	45.7796	9367.0000	0.9137	0.0187	0.9608	0.0045	0.6977	0.0491	0.8481	0.0150
60	resnet50_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	52.6868	9367.0000	0.9199	0.0234	0.9633	0.0052	0.7029	0.0592	0.8599	0.0110
61	resnet101_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.1196	9367.0000	0.9289	0.0140	0.9635	0.0042	0.7120	0.0464	0.8587	0.0128
62	resnet152_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	62.0644	9413.0000	0.9327	0.0188	0.9668	0.0050	0.7151	0.0499	0.8604	0.0132
63	squeezenet1_0_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	43.7105	9417.0000	0.9186	0.0186	0.9701	0.0056	0.7185	0.0518	0.8879	0.0118
64	squeezenet1_1_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	41.6251	9419.0000	0.9087	0.0288	0.9691	0.0053	0.7139	0.0570	0.8896	0.0118
65	densenet121_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	53.7737	9451.0000	0.9275	0.0195	0.9673	0.0064	0.7136	0.0474	0.8726	0.0147
66	densenet169_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	53.0848	9503.0000	0.9260	0.0166	0.9678	0.0048	0.7148	0.0563	0.8804	0.0131
67	densenet161_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	60.5094	9559.0000	0.9325	0.0160	0.9721	0.0046	0.7127	0.0481	0.8891	0.0127
68	densenet201_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	54.9551	9617.0000	0.9262	0.0171	0.9708	0.0054	0.7037	0.0501	0.8803	0.0124
69	inception_v3_torchvision	imagenet	1,000	299	[0.485 0.456 0.406]	[0.229 0.224 0.225]	60.5731	9641.0000	0.9127	0.0205	0.9681	0.0050	0.6995	0.0437	0.8870	0.0127
70	googlenet_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	43.8201	9653.0000	0.9252	0.0222	0.9661	0.0055	0.7142	0.0489	0.8746	0.0108

Journal of Geophysical Research: Space Physics

10.1029/2021JA029683

Table A1
Continued

	Model	Key	num_classes	Size	Mean	Std	Diff	Mem	acc_test_class2	dev_test_class2	acc_train_class2	dev_train_class2	acc_test_class6	dev_test_class6	acc_train_class6	dev_train_class6
71	shufflenet_v2_x0_5_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	39.8624	9653.0000	0.9117	0.0218	0.9706	0.0060	0.7069	0.0458	0.8968	0.0092
72	shufflenet_v2_x1_0_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	40.6544	9655.0000	0.9177	0.0251	0.9732	0.0060	0.7094	0.0472	0.9086	0.0107
73	mobilenet_v2_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	44.5089	9663.0000	0.9150	0.0196	0.9617	0.0051	0.7046	0.0447	0.8635	0.0121
74	resnext50_32x4d_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	57.5930	9671.0000	0.9204	0.0189	0.9655	0.0055	0.7159	0.0454	0.8645	0.0112
75	resnext101_32x8d_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	76.8500	9683.0000	0.9245	0.0179	0.9681	0.0051	0.7192	0.0508	0.8724	0.0116
76	wide_resnet50_2_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.4168	9691.0000	0.9197	0.0229	0.9655	0.0046	0.7060	0.0566	0.8646	0.0141
77	wide_resnet101_2_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	65.0305	9701.0000	0.9212	0.0168	0.9626	0.0046	0.6964	0.0508	0.8617	0.0145
78	mnasnet0_5_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	42.2927	9703.0000	0.9180	0.0210	0.9695	0.0054	0.7059	0.0424	0.8891	0.0117
79	mnasnet1_0_torchvision	imagenet	1,000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	47.8704	9713.0000	0.9215	0.0226	0.9729	0.0043	0.7240	0.0479	0.8941	0.0122

Note. The "model" column contains the networks' identifier, Where the first part is the name, the second part the source of the network, the "key" on which set of data it has been trained on, "num_class" the amount of features extracted by the network, "size", "mean" and "std" the required size and normalization values for the provided image, "diff" and "mem" the time in seconds and used memory in MB to extract all features, "acc_test_class2", "dev_test_class2", "acc_train_class2", "dev_train_class2", "acc_test_class6", "dev_test_class6", "acc_train_class6", "dev_train_class6" the accuracies and standard deviations for test and train set in different classes when predicting OATH Images. The table is sorted in the order of which the neural networks were evaluated. floating point numbers have been rounded to 4 decimal places for convenience. The table is accessible in machine readable form as the "times.csv"-file.

the amount of output classes, equal to the amount of classes in the traning dataset. “size” is the size of the input image the neural network requires. This size has to be matched exactly, images therefore might need to be scaled, resized, cropped or padded to match this requirement. “mean” and “std” are the mean and standard deviation of the training dataset in the RGB-channels of the image. All images have to be transformed according to these values. “diff” and “mem” are the time used for extraction of features from all images in the OATH-dataset and the amount of memory used during extraction. The last four columns give the accuracy and standard deviation for 2 and 6 classes prediction each.

Appendix B: Gridsearch for Hyperparameters

Figure B1 shows how the hyperparameters influence the accuracy achieved by an SVM with linear kernel. For $C = 10^{-3.8} \approx 1.58{*}10^{-4}$ we see the highest accuracy of 73.00 ± 5.22 for six-class and 91.27 ± 2.83 for two-class classification.

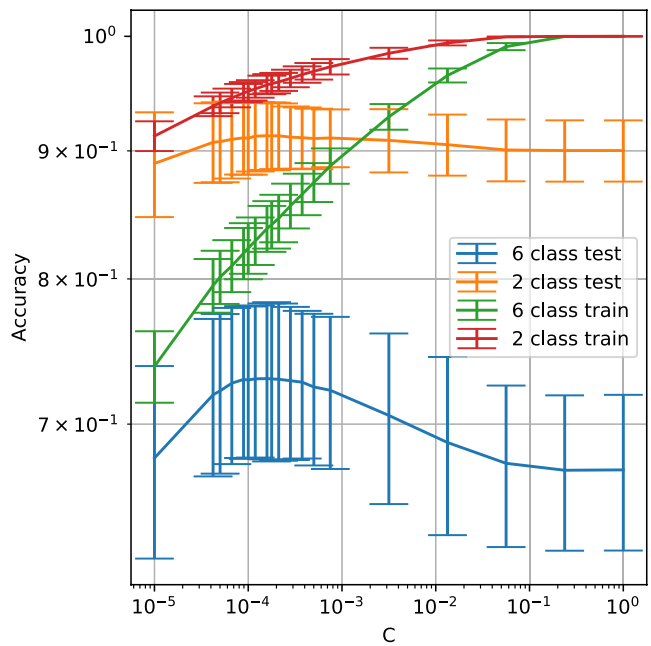

Figure B1. Accuracy achieved for a linear SVM for different hyperparameters C.

In Figure B2 we see the training and testing accuracies for 2-class and 6-class classification for different hyperparameters on an SVM with RBF-kernel trained on the features extracted by the ShuffleNetV2 network. To emphasize the highest values, we have cut off values below 85% for 2-class and 67% for 6-class classification. We see that the highest accuracy of 73.03 ± 5.47% for six-class and 91.03 ± 2.82 for two-class classification is achieved for multiple sets of hyperparameters. These values are not higher than those obtained by the linear SVM. The linear SVM is however considerably faster to train and less prone to overfitting for such a low relation of training values to feature space dimensionality.

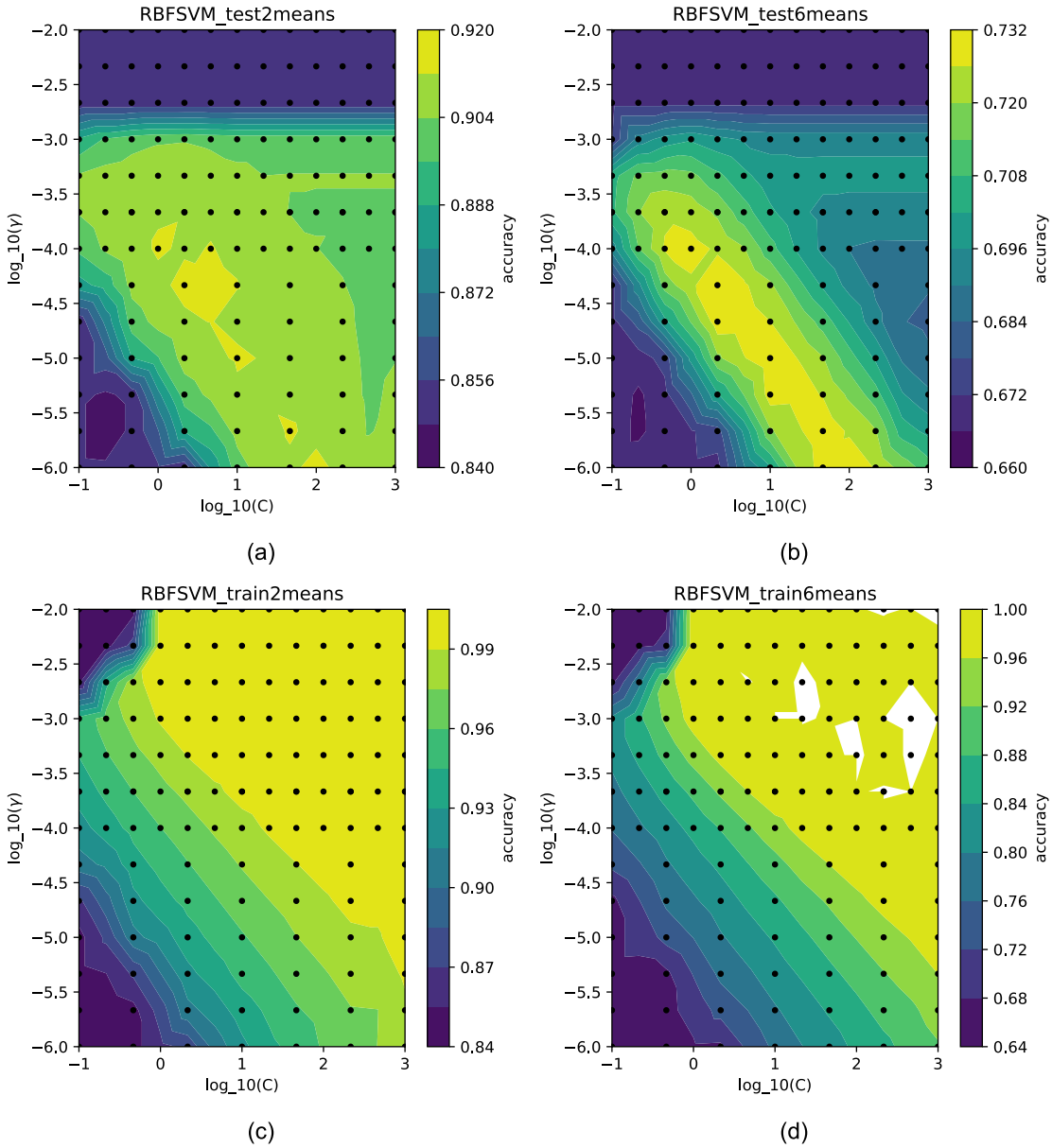

Figure B2. Accuracy achieved for an SVM with RBF-kernel for different hyperparameters γ and C. The top row shows the average performance achieved by the testing splits and the bottom row by the training splits. The left column shows accuracy for 2-class classification and the right column for 6-class.

Appendix C: Additional Figures

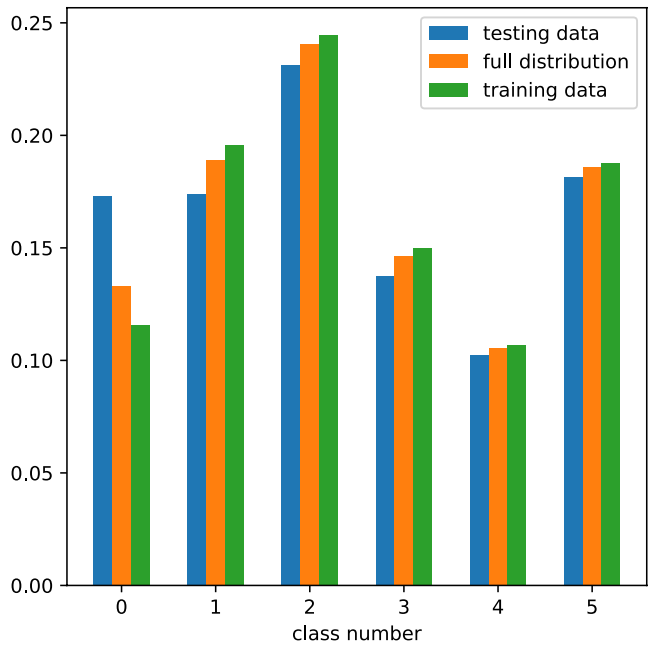

Figure C1. Distribution of the classes in the training and testing dataset compared to the distribution of classes in the whole dataset when training the final all sky image classifier.

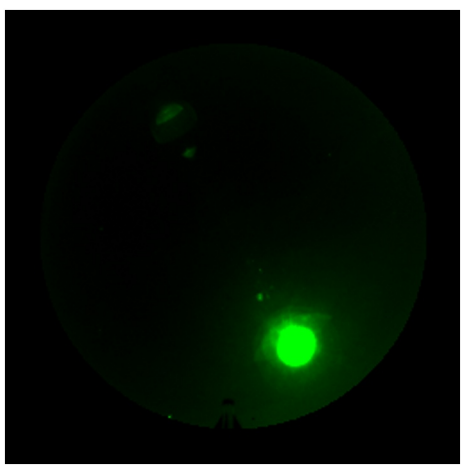

Figure C2. OATH Image 00568, an example of an image showing the moon with lens flares and other types of reflections.

Data Availability Statement

We thank the University of Oslo, TGO, IMAGE and AWI for publishing their research data and enabling this work. The All Sky Imager Data and OATH Dataset used in this work is available through the University of Oslo (http://tid.uio.no/plasma/aurora/ and http://tid.uio.no/plasma/oath/), the Magnetometer Data through IMAGE (https://space.fmi.fi/image/www/index.php) and the ceilometer data through AWIPEV (https://doi.org/10.1594/PANGAEA.880300). We provide the data and code supporting TAME openly and freely on http://tid.uio.no/TAME/ and https://doi.org/10.11582/2021.00071 and encourage anyone to use our classifier.

Acknowledgments
This work is a part of the 4DSpace Strategic Research Initiative at the Department of Physics, University of Oslo. WJM acknowledges funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC Consolidator Grant agreement No. 866357, POLAR-4DSpace).

References

Akasofu, S.-I. (1964). The development of the auroral substorm. *Planetary and Space Science*, *12*(4), 273–282. https://doi.org/10.1016/0032-0633(64)90151-5

Cadene, R. (2020). *pretrained-models. pytorch*. Retrieved from https://github.com/Cadene/pretrained-models.pytorch/tree/8aae3d8f1135b6b13fed79c1d431e3449fdbf6e0. GitHub

Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., & Feng, J. (2017). *Dual path networks*. arXiv preprint arXiv:1707.01629.

Clausen, L. B. N., & Nickisch, H. (2018). Automatic classification of auroral images from the Oslo auroral THEMIS (OATH) data set using machine learning. *Journal of Geophysical Research: Space Physics*, *123*(7), 5640–5647. https://doi.org/10.1029/2018JA025274

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer Science & Business Media.

Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2020). Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(8), 2011–2023. https://doi.org/10.1109/TPAMI.2019.291337210.1109/TPAMI.2019.2913372

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2261–2269). https://doi.org/10.1109/CVPR.2017.243

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/306538610.1145/3065386

Kvammen, A., Wickstrøm, K., McKay, D., & Partamies, N. (2020). Auroral image classification with deep neural networks. *Journal of Geophysical Research: Space Physics*, *125*(10), e2020JA027808. https://doi.org/10.1029/2020ja027808

Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer vision – ECCV 2018* (pp. 122–138). Springer International Publishing. https://doi.org/10.1007/978-3-030-01264-9_8

Maturilli, M., & Ebell, K. (2018). Twenty-five years of cloud base height measurements by ceilometer in Ny-ålesund, Svalbard. *Earth System Science Data*, *10*(3), 1451–1456. https://doi.org/10.5194/essd-10-1451-2018

Maturilli, M., & Herber, A. (2017). Ceilometer cloud base height from station Ny-Ålesund from August 1992 to July 2017, reference list of 290 datasets [data set]. *PANGAEA*. https://doi.org/10.1594/PANGAEA.880300 (Supplement to: Maturilli, Marion; Ebell, Kerstin (2018): Twenty-five years of cloud base height measurements by ceilometer in Ny-Ålesund, Svalbard. Earth System Science Data, 10(3), 1451–1456, https://doi.org/10.5194/essd-10-1451-2018

McGranaghan, R. M., Ziegler, J., Bloch, T., Hatch, S., Camporeale, E., Lynch, K., et al. (2021). Toward a next generation particle precipitation model: Mesoscale prediction through machine learning (a case study and framework for progress). *Space Weather*, *19*(6), e2020SW002684. https://doi.org/10.1029/2020sw002684

McInnes, L., Healy, J., Saul, N., & Gro berger, L. (2018). UMAP: Uniform Manifold approximation and projection. *Journal of Open Source Software*, *3*(29), 861. https://doi.org/10.21105/joss.0086110.21105/joss.00861

McKay, D., & Kvammen, A. (2020). Auroral classification ergonomics and the implications for machine learning. *Geoscientific Instrumentation, Methods and Data Systems*, *9*(2), 267–273. https://doi.org/10.5194/gi-9-267-2020

McPherron, R. L., Aubry, M. P., Russell, C. T., & Coleman, P. J., Jr (1973). Satellite studies of magnetospheric substorms on august 15, 1968: 4. Ogo 5 magnetic field observations. *Journal of Geophysical Research (1896-1977)*, *78*(16), 3068–3078. https://doi.org/10.1029/ja078i016p03068

Mende, S. B., Harris, S. E., Frey, H. U., Angelopoulos, V., Russell, C. T., Donovan, E., et al. (2009). The THEMIS array of ground-based observatories for the study of auroral substorms. In J. L. Burch, & V. Angelopoulos (Eds.), *The themis mission* (pp. 357–387). Springer. https://doi.org/10.1007/978-0-387-89820-9_1610.1007/978-0-387-89820-9_16

Murphy, K. R., Mann, I. R., Rae, I. J., Waters, C. L., Frey, H. U., Kale, A., et al. (2013). The detailed spatial structure of field-aligned currents comprising the substorm current wedge. *Journal of Geophysical Research: Space Physics*, *118*(12), 7714–7727. https://doi.org/10.1002/2013ja018979

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.. Retrieved from https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

Rae, I., Murphy, K., Watt, C. E., Mann, I. R., Yao, Z., Kalmoni, N. M., et al. (2017). Using ultra-low frequency waves and their characteristics to diagnose key physics of substorm onset. *Geoscience letters*, *4*(1), 1–11. https://doi.org/10.1186/s40562-017-0089-0

Raschka, S. (2015). *Python machine learning*. Packt publishing ltd.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Y. Bengio, & Y. LeCun (Eds.), *3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings*. Retrieved from http://arxiv.org/abs/1409.1556

Syrjäsuo, M., & Donovan, E. (2002). Analysis of auroral images: Detection and tracking. *Geophysica*, *38*(1–2), 3–14.

Syrjäsuo, M., Donovan, E., & Peura, M. (2002). Using attribute trees to analyse auroral appearance over Canada. In *Sixth IEEE workshop on applications of computer vision 2002 (WACV 2002) proceedings*. (pp. 289–295). https://doi.org/10.1109/ACV.2002.1182196

Syrjäsuo, M., Donovan, E., Qin, X., & Yang, Y. (2007). Automatic classification of auroral images in substorm studies. In *8th International conference on substorms (ICS8)* (pp. 309–313).

Syrjäsuo, M., Kauristie, K., & Pulkkinen, T. (2001). A search engine for auroral forms. *Advances in Space Research*, *28*(11), 1611–1616. https://doi.org/10.1016/s0273-1177(01)00492-6

Syrjäsuo, M., & Pulkkinen, T. (1999). Determining the skeletons of the auroras. In *Proceedings 10th International conference on image analysis and processing* (pp. 1063–1066). https://doi.org/10.1109/ICIAP.1999.797739

Syrjäsuo, M. T., & Donovan, E. F. (2004). Diurnal auroral occurrence statistics obtained via machine vision. *Annales Geophysicae*, *22*(4), 1103–1113. https://doi.org/10.5194/angeo-22-1103-2004

Syrjäsuo, M. T., & Donovan, E. F. (2005). Using relevance feedback in retrieving auroral images. In *Computational intelligence* (pp. 420–425).

Syrjäsuo, M. T., Donovan, E. F., & Cogger, L. L. (2004). Content-based retrieval of auroral images-thousands of irregular shapes. *Proceedings of the Fourth IASTED Visualization, Imaging, and Image Processing*, (pp. 224–228).

Syrjäsuo, M. T., Kauristie, K., & Pulkkinen, T. I. (2000). Searching for aurora. In *Proceedings of the IASTED International conference on signal and image processing, sip* (pp. 381–386).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-ResNet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/11231

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2818–2826). https://doi.org/10.1109/CVPR.2016.308

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., et al. (2019). MnasNet: Platform-Aware neural architecture search for mobile. In *2019 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2815–2823). https://doi.org/10.1109/CVPR.2019.00293

Tanskanen, E. I. (2009). A comprehensive high-throughput analysis of substorms observed by IMAGE magnetometer network: Years 1993–2003 examined. *Journal of Geophysical Research*, *114*(A5). https://doi.org/10.1029/2008ja013682

Wang, L. (2005). *Support vector machines: Theory and applications* (Vol. 177). Springer Science & Business Media.

Yang, Q., Tao, D., Han, D., & Liang, J. (2019). Extracting auroral key local structures from all-sky auroral images by artificial intelligence technique. *Journal of Geophysical Research: Space Physics*, *124*(5), 3512–3521. https://doi.org/10.1029/2018ja026119

Yang, X., Gao, X., Song, B., & Yang, D. (2018). Aurora image search with contextual CNN feature. *Neurocomputing*, *281*, 67–77. https://doi.org/10.1016/j.neucom.2017.11.059

Yang, X., Wang, N., Song, B., & Gao, X. (2019). BoSR: A CNN-based aurora image retrieval method. *Neural Networks*, *116*, 188–197. https://doi.org/10.1016/j.neunet.2019.04.012

Zhang, X., Li, Z., Loy, C. C., & Lin, D. (2017). Polynet: A pursuit of structural diversity in very deep networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3900–3908). https://doi.org/10.1109/CVPR.2017.415