Transfer Learning Aurora Image Classification and Magnetic Disturbance Evaluation

Pascal Sado[1], Lasse Boy Novock Clausen[1], Wojciech Jacek Miloch[1], and Hannes Nickisch[2]

[1]University of Oslo
[2]Philips Research

November 30, 2022

Abstract

We develop an open source algorithm to apply Transfer learning to Aurora image classification and Magnetic disturbance Evaluation (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 running time and feature accuracy. 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 90% accuracy in the six classes; if we aggregate the 3 auroral and 3 non-aurora classes after classification, we achieve up to 98% 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 provided, which opens possibility for large scale studies of all-sky images.

Transfer Learning Aurora Image Classification and Magnetic Disturbance Evaluation

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

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

Key Points:

- A pretrained CNN feature extractor and an SVM can successfully classify all-sky images
- Up to 98% accuracy can be achieved in the classification
- From the underlying image feature representation we predict physical quantities such as magnetic disturbance and cloud height

Corresponding author: Pascal Sado, `Pascal.Sado@fys.uio.no`

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 running time and feature accuracy. 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 90% accuracy in the six classes; if we aggregate the 3 auroral and 3 non-aurora classes after classification, we achieve up to 98% 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 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 98% 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 behaviours simultaneously across the polar region has been first shown in 1964, when analysing 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 towards 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 the most important tool to analyse 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, the imagers operated by the University of Oslo (UiO) have taken approximately 8 million images. THEMIS is long term operating using more cameras with about a hundred million images acquired, both imager arrays adding several hundred thousand images each season. Analysing and labelling images manually and consistently would take many humans to even keep up with the production of images, not including the aforementioned backlog, which is why researchers usually select smaller events on the timescale of a few hours or days and manually analyse the images in combination with other sources (see for example (Murphy et al., 2013; Rae et al., 2017)). It is clear that statistical analyses at scale require highly automatic image processing algorithms.

First attempts have been performed by M. Syrjäsuo and Pulkkinen (1999) who determined the skeletons of auroral images which they later used to find images showing auroral arcs (M. T. Syrjäsuo et al., 2000; M. Syrjäsuo et al., 2001). Later M. 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 (M. Syrjäsuo et al., 2002; M. T. Syrjäsuo & Donovan, 2004) for example by using Fourier descriptors on a segmented image (M. T. 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 (M. T. Syrjäsuo & Donovan, 2005) led to an automated classifier (M. Syrjäsuo et al., 2007) making use of Basic Gray Level Aura Matrices (BGLAM) to classify images into 6 classes (unknown, cloudy and 4 types of aurora) with 70-80% accuracy.

Another approach is to use convolutional neural networks as aides in classifying and categorizing 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 labelled images in 1000 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 stepping stone, 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, X. Yang et al. (2018) finetuned the pretrained network, 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 (X. Yang 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% (Q. Yang 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 7 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.
Using Generative adversarial networks (GANs), a special form of unsupervised neural networks, to extract Key Local Structures around the aurora also allows for retrieval of similar aurora images with an accuracy of up to 92% (Q. Yang et al., 2019).
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 smaller or simpler models at greatly reduced training cost. This method has already been shown to be successful by Clausen and Nickisch (2018), who achieved 82% accuracy distinguishing aurora images between 6 classes - 3 different classes of aurora and 3 different non-aurora classes.
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 98% aggregated classification accuracy when distinguishing between images that show aurora and images that do not show aurora and 90% 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 [1] and can be used to classify vast amounts of all sky images in a short amount of time.

2 Description of Data Sources

We combine data from 3 different sources.
The first is an All Sky Imager, taking images of the sky when the Sun is below the horizon in intervals of 10 s to 60 s in wavelengths of 5570 Å and 6300 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 narrow monochromatic filter allowing only the desired wavelengths to pass, taking images through a fisheye lense with a field of view of 180°. The camera saturates at a luminosity of ≈9 kR and produces images at a resolution of 460 px by 460 px. The images are stored as 16-bit gray level files, where the luminosity is roughly linear to the brightness of the pixel. In order to protect the camera from oversaturation 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) (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.
At last 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.

[1] `https://creativecommons.org/licenses/by/4.0/`

3 Methods

3.1 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 labelled 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.
For our purposes, we remove the last layer of the neural network - the classification layer - and are left with a 1000 dimensional feature vector $\vec{\varphi}_i = \vec{\varphi}(x_i) = (\varphi_{i,0}, \varphi_{i,1}, \varphi_{i,2}, ..., \varphi_{i,999})$ 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.
We are using a pretrained neural network instead of training one ourselves or finetuning this existing neural network to our liking because we want to evaluate possible out-of-the-box solutions that can be achieved using readily available tools in just a few lines of code and without the need of computationally expensive methods.
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.

3.2 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). 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 target label $t \in 0, 1$ and a predicted function value $f \in \mathbb{R}$

$$\begin{aligned} \ell_2(t, f) &= (t - f)^2, \\ \ell_h(t, f) &= \max(1 - tf), \\ \ell_c(t, f) &= -t \log f - (1 - t) \log(1 - f) \end{aligned} \tag{1}$$

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 regularised objective function becomes

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

where N is the amount of samples, y_i the target value, $\vec{w}$ the vector of weights and $\vec{\varphi}_i$ the vector of features belonging to the image and training value. Depending on the regularization parameter λ, the additional term $\lambda ||\vec{w}||_2^2$ 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 $\lambda = 1/(2CN)$ 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

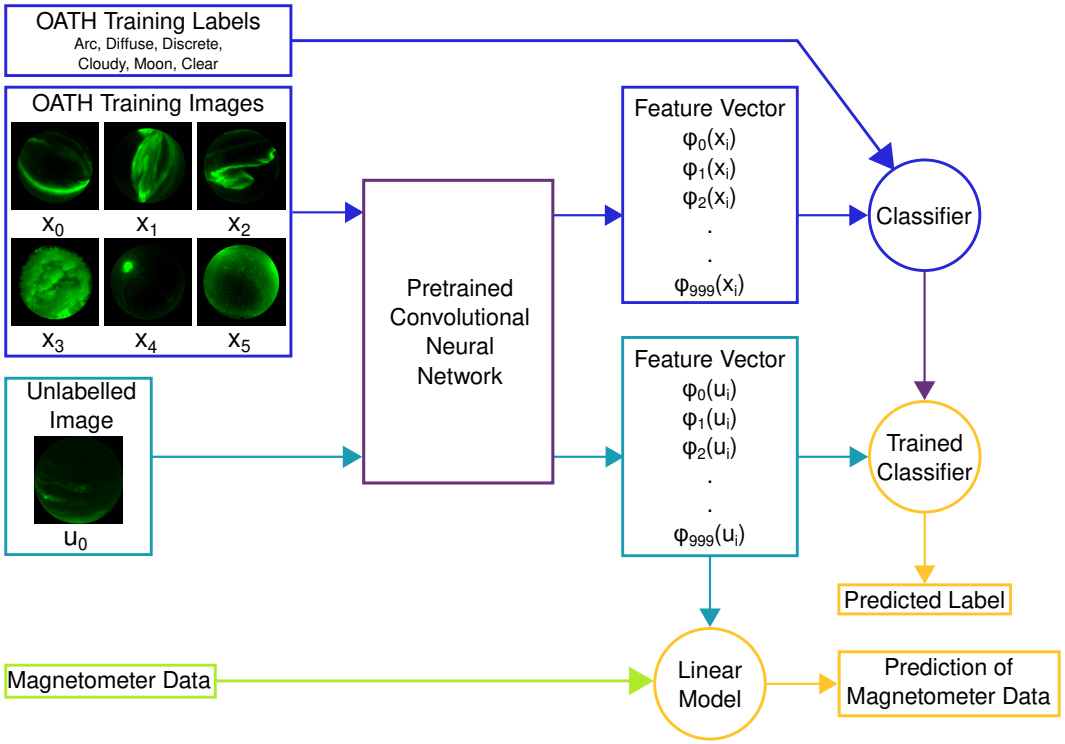


Figure 1: The flow of data (images and magnetometer) used in this work

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.

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

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 optimised in closed form by solving a linear system, while the SVM requires a numerical approach to optimise its convex but non-linear hinge loss function $\ell_h(t, f)$ (Hastie et al., 2009).
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).

4 Results

4.1 Creating a new classifier

To find the best pretrained neural network suitable for our work, we evaluated several network architectures compatible with 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 features are also more space efficient and therefore easier to access and use for later processing. One TB of images will be reduced to features the size of a few GB, 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"[2], 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 normalized 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 to match this requirement. Most neural networks we tested required an input size of $224\,\mathrm{px} \times 224\,\mathrm{px}$, about half the size of our original images of $460\,\mathrm{px} \times 460\,\mathrm{px}$. We provide a full list of all tested neural networks including the used parameters in the appendix in Table A1.

In Table 1 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 2.

Table 1: 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. 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. Networks denoted by † have been installed through "pretrained-models.pytorch" (Cadene, 2020), those denoted by ‡ through torchvision.

model name	time [s]	images processed per second	2 class acc [%]	6 class acc [%]	2 class acc ranking	6 class acc ranking
SE-ResNet152 †(Hu et al., 2020)	67.71	86	95.98 ± 0.31	83.66 ± 0.48	8	1
SE-ResNet50 †(Hu et al., 2020)	50.68	115	96.06 ± 0.28	83.52 ± 0.58	6	2
SENet154 †(Hu et al., 2020)	101.32	57	95.76 ± 0.44	83.15 ± 0.56	16	3
ShuffleNet V2 x1_0 ‡(Ma et al., 2018)	40.65	143	95.51 ± 0.29	83.05 ± 0.44	28	4
MNASNet 1_0 ‡(Tan et al., 2019)	47.87	122	96.68 ± 0.37	82.95 ± 0.79	1	5
DenseNet161 †(Cadene, 2020)	64.70	90	95.83 ± 0.27	82.83 ± 0.50	10	6
SE-ResNet101 †(Hu et al., 2020)	57.30	102	96.50 ± 0.45	82.81 ± 0.88	2	7
DenseNet161 ‡(Huang et al., 2017)	60.50	96	96.07 ± 0.49	82.47 ± 1.21	5	10
SE-ResNeXt101_32x4d †(Hu et al., 2020)	66.58	87	96.29 ± 0.31	82.47 ± 0.64	3	11
MNASNet 0_5 ‡(Tan et al., 2019)	42.29	138	95.54 ± 0.46	82.30 ± 0.78	25	13
SE-ResNeXt50_32x4d †(Hu et al., 2020)	55.15	106	95.61 ± 0.34	82.27 ± 0.44	24	14
InceptionV3 †(Szegedy et al., 2016)	59.16	98	95.47 ± 0.22	82.18 ± 0.26	30	15
InceptionV3 ‡(Szegedy et al., 2016)	60.57	96	95.14 ± 0.29	80.66 ± 1.07	42	40
InceptionV4 †(Szegedy et al., 2017)	73.14	80	95.04 ± 0.37	79.69 ± 0.71	46	53
VGG ‡(Simonyan & Zisserman, 2015)	56.42	103	93.82 ± 0.20	77.49 ± 0.34	74	75

[2] `https://pytorch.org/vision/stable/index.html`

Table 2: The different class labels and their explanations

Label	Explanation	Class 2	Class 6
arc	The image shows aurora with well defined edges with 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

For each set of extracted features, the data was split into 5 randomly selected splits of 70% training and 30% testing data. The mean accuracy and standard deviation of the 5 trained and predicted runs is given as the accuracy and standard deviation in the table.
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 SE-ResNet152 for example performed best in classifying the images in six classes and they were the eight best features in classifying the images in 2 classes. 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. The fourth best performing model "ShuffleNet V2" (Ma et al., 2018) is however significantly faster than other networks. It was able to extract features from all 5824 images in 41 s and classified $95.51 \pm 0.29\%$ of the two-class labelled images and $83.05 \pm 0.44\%$ of the six-class labeled images correctly. 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 trained an SVM with RBF kernel on the extracted features. Using tenfold cross-validation we perform a grid search for the optimal hyperparameters (cf. Figure B1 in the appendix). For $\gamma = 0.001$ and $C = 10$ the classifier performs best and achieves an accuracy of $89.58 \pm 0.90\%$ in six class classification. Classifying the testing data we observe the following confusion matrix for the last of the 10 runs:

	arc	*diffuse*	*discrete*	*cloud*	*moon*	*clear*
arc	166	13	28	0	0	4
diffuse	15	277	39	0	0	8
discrete	22	23	383	1	1	5
cloud	0	2	1	249	4	0
moon	0	0	0	2	178	2
clear	7	12	4	0	1	300

We see that the biggest confusion arises between the different classes of northern lights. If we bundle this into two classes - aurora and no aurora - we achieve an accuracy of $97.81 \pm 0.41\%$. Another issue is that northern lights images are classified as images showing clear skies and vice versa. This might be because some images show northern lights which are very faint and humans as well as the classifier cannot decide consistently whether to classify them as aurora or not.
Overall there is a significant improvement over the classifier previously demonstrated by Clausen and Nickisch (2018) which achieved an accuracy of $81.7 \pm 0.1\%$.

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.
We use the previously mentioned timeframe to show some examples and test how well the classifier has performed.
Figure 2 shows the 6 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 slanted line is drawn at a probability of 20%, 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 2.
For the images classified as arcs (Figure 2a) we see different images of arcs. The second and third image in the right show only faint aurora, but the visible features still span the whole image in a single bow. The other images show a much brighter, discrete, single bow of aurora spanning over the whole image. Although arcs often follow an east-west-direction, the classifier also managed to classify the north-south images correctly. This is most likely due to the fact that the training images were randomly rotated before the features were extracted for training the classifier. The classifier therefore has no bias towards rotation of the images.
Figure 2b shows the diffuse auroras and we can see that the images represent this class well. While the aurora are clearly visible, there is no clear structure in them.
Figure 2c shows the discrete, but not arc-like aurora. Every image shows one or more warped or deformed arcs, but no arcs that span in a single, well-defined bow over the whole image.
Figure 2d corresponds to a cloudy image. 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 2e was to label images as showing the moon. To avoid damage due to 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 48 images out of the subset of 550286 of the 2010/2011 season in total were labelled 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. The second image for example was taken on midnight of New Year's Eve and shows exploding fireworks.
Figure 2f shows a clear night sky with some auroral activity near the bottom left field of view of the imager or a slightly cloudy image with auroral activity in the background. These images would have likely been rejected by a human because they do not show enough and clear aurora to be useful.
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, the classifier is not very confident in them and they sometimes show phenomena in the image the classifier has never seen before. The classifier is therefore very confident in

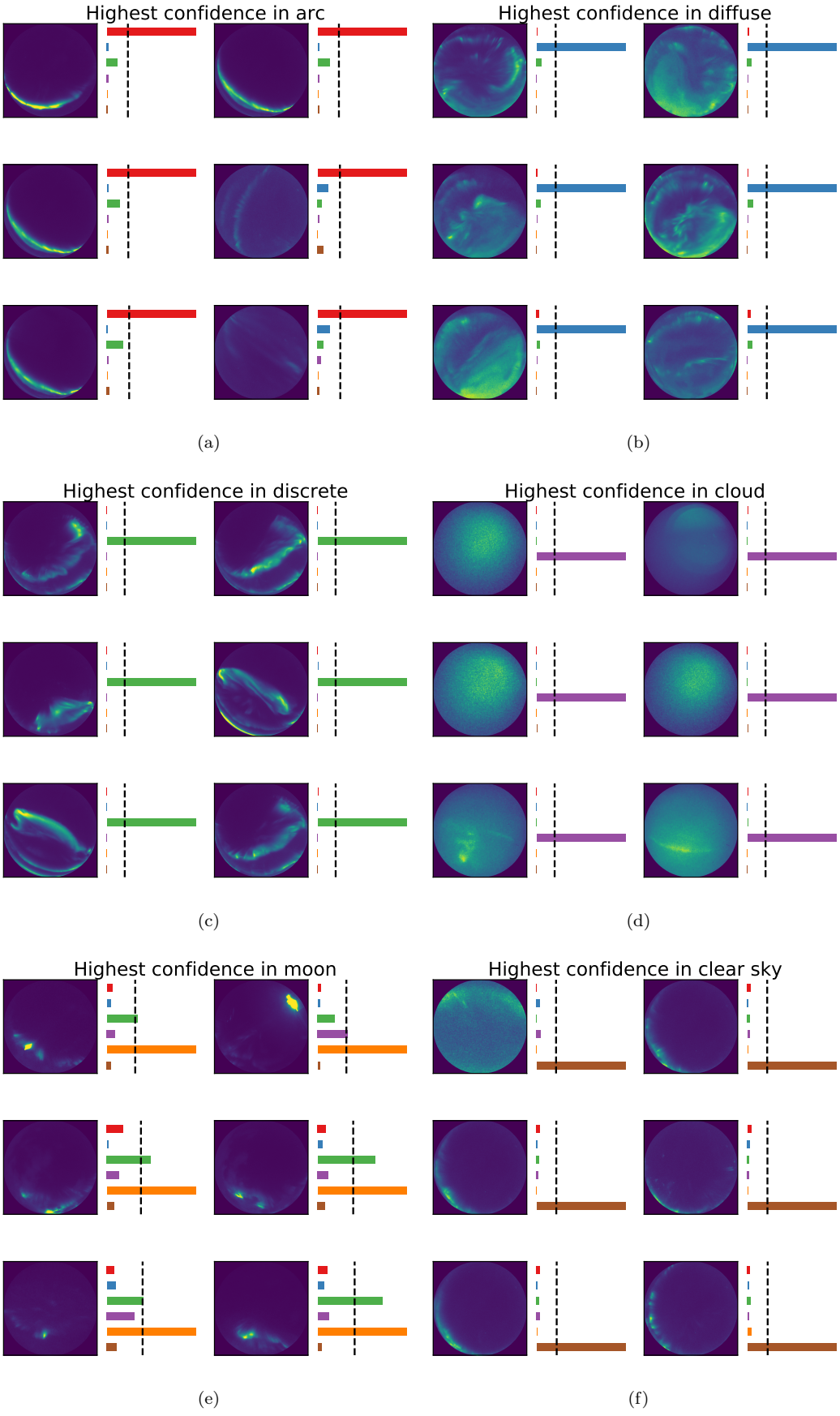
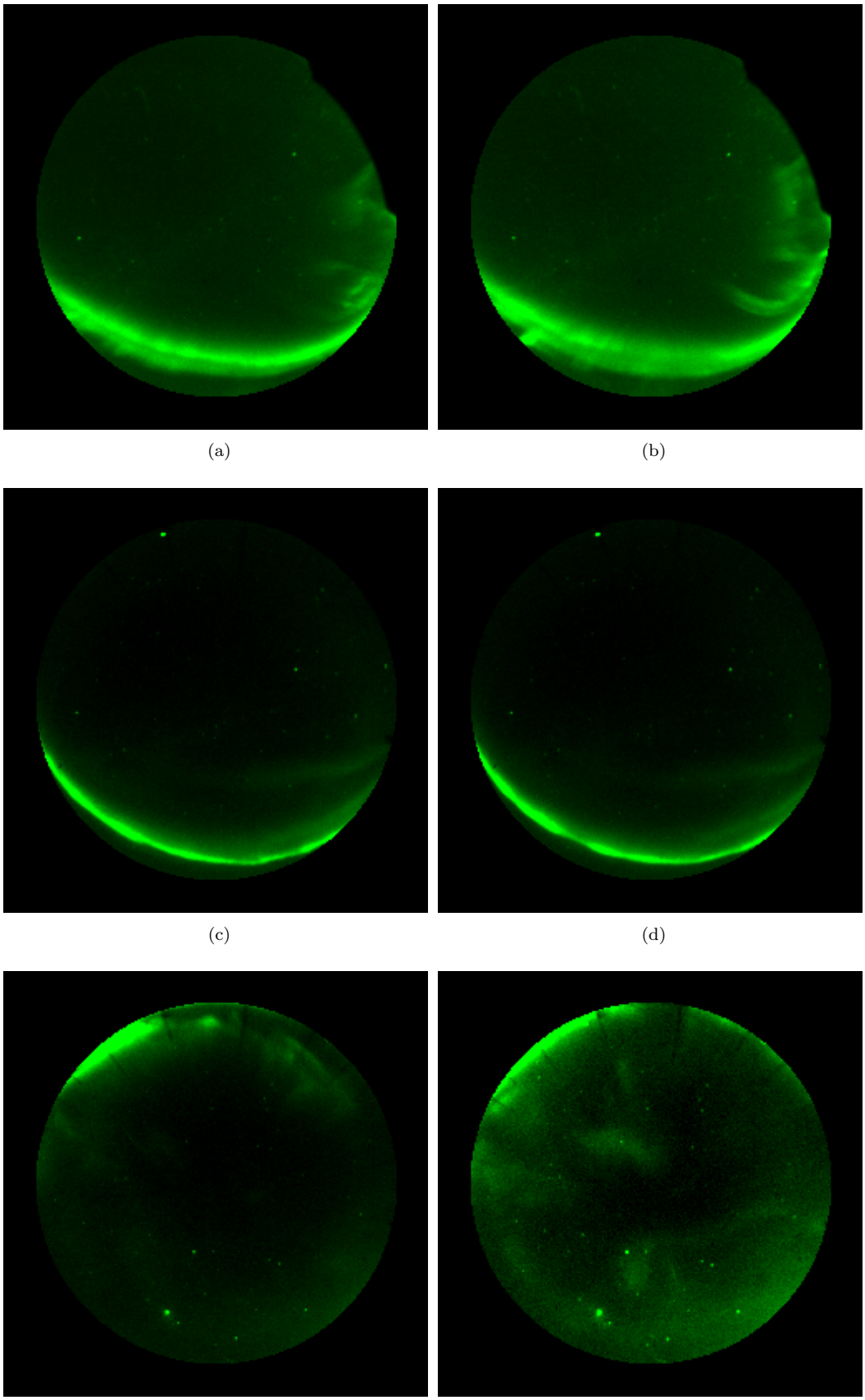


Figure 2: Images with highest confidences for each class (a) arcs, (b) diffuse aurora, (c) discrete aurora, (d) cloudy, (e) moon (f) clear sky. The probability assigned to each class are given as bar graph normalized to the highest probability for each image beside each image. The dashed lines indicate equal class probability of 20%. The colors of the bars correspond to the colors of the classes shown in Table 2.

images that would have been labelled with the same confidence by a human operator. We have however seen above that sometimes there are difficulties for the classifier to distinguish between aurora classes and classification as clear skies.
In Figure 3 we see an explanation why the classifier might have problems discerning between 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. Images like these were probably labelled as "clear skies" because they are of no scientific use. These borderline cases are already ambiguous when labelled by a human, because different people would draw the line between usable and not-usable somewhere else.
The same way we analyzed the images the classifier had its highest confidence in, we can analyze the images with the lowest confidence. These are the images where the label is assigned but the probability of the image being in the class is lowest for all images assigned to this class. We have plotted the images in Figure 4, the style is the same as for Figure 2. For the three aurora classes and the clear sky class we see that there are mostly images with auroral features near the bottom border of the image on an otherwise clear sky. These are again evidence for the problem of the classifier to distinguish edge cases.
For the third class (cloudy) we see three of those images and two images that show a mostly dark background without any stars visible and only very faint aurora. For all of these the probabilities are again 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.
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 testing data performs well with 89.6% six-class accuracy. We show that our classifier handles unknown images from a different source equally well. In both cases, misclassification is largest for cases that would also be ambiguous to humans.

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 5 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). 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 1000-dimensional feature space condensed into a 2D space. All 550286 images have been used to create the map and a random selection of 3000 images has been used to create the figure. Each point in the plot represents one of the 3000 selected images. The colors represent the classes they belong to, using the same color scheme as explained in Table 2 and used in Figures 2 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 optically similar images close to each other and optically 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 map-



(a) (b)

(c) (d)

(e) (f)

Figure 3: Labelled images of the OATH dataset. Left column: images labelled as "clear skies / no aurora". Right column: images labelled "discrete", "arc", "diffuse" respectively

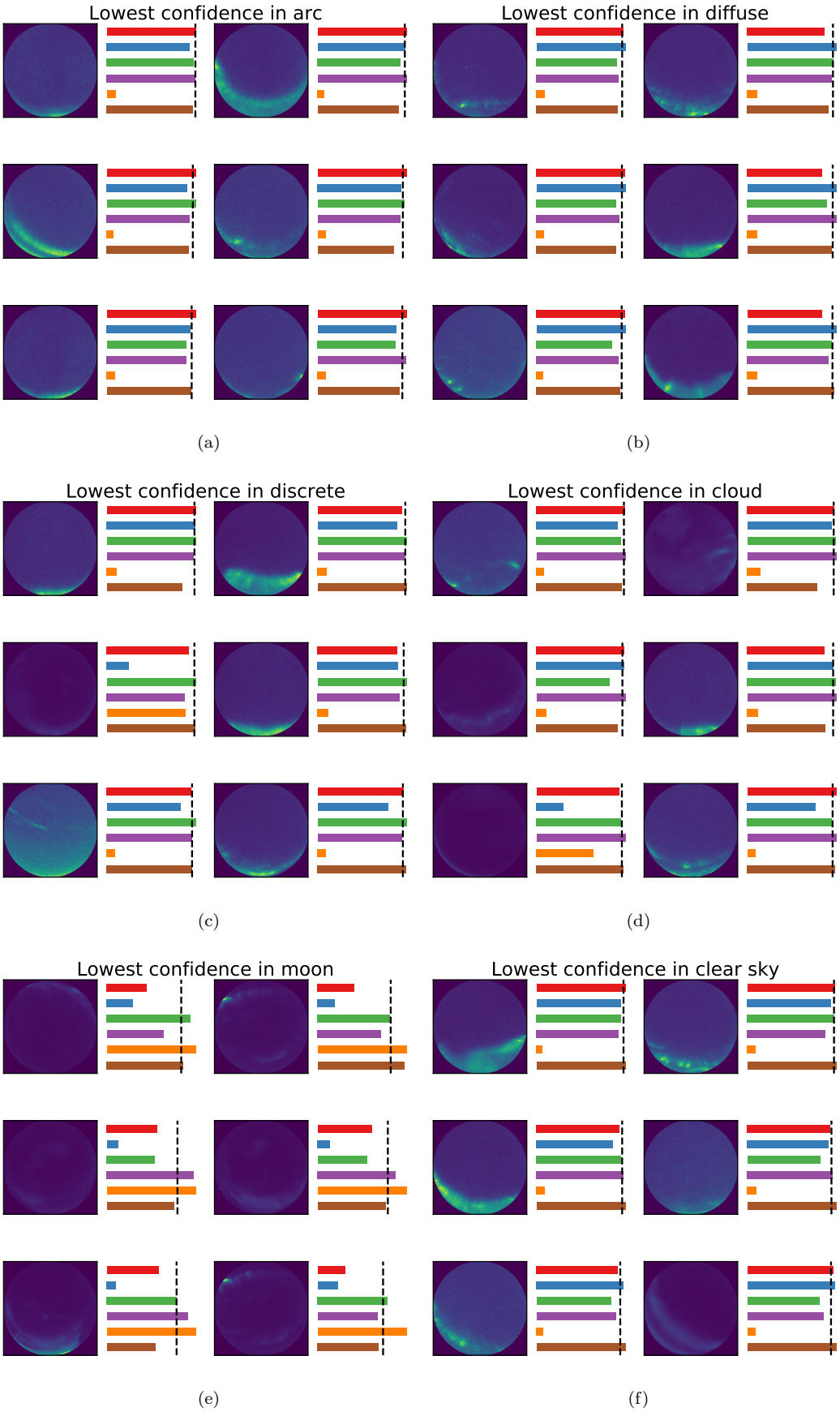


Figure 4: Images with lowest confidences for each class (a) class 0, (b) class 1, (c) class 2, (d) class 3, (e) class 4 (f) class 5. The probability assigned to the label by the classifier is given below each individual image. The slanted lines give a probability of 20%. The colors of the bars correspond to the colors of the classes shown in Table 2.

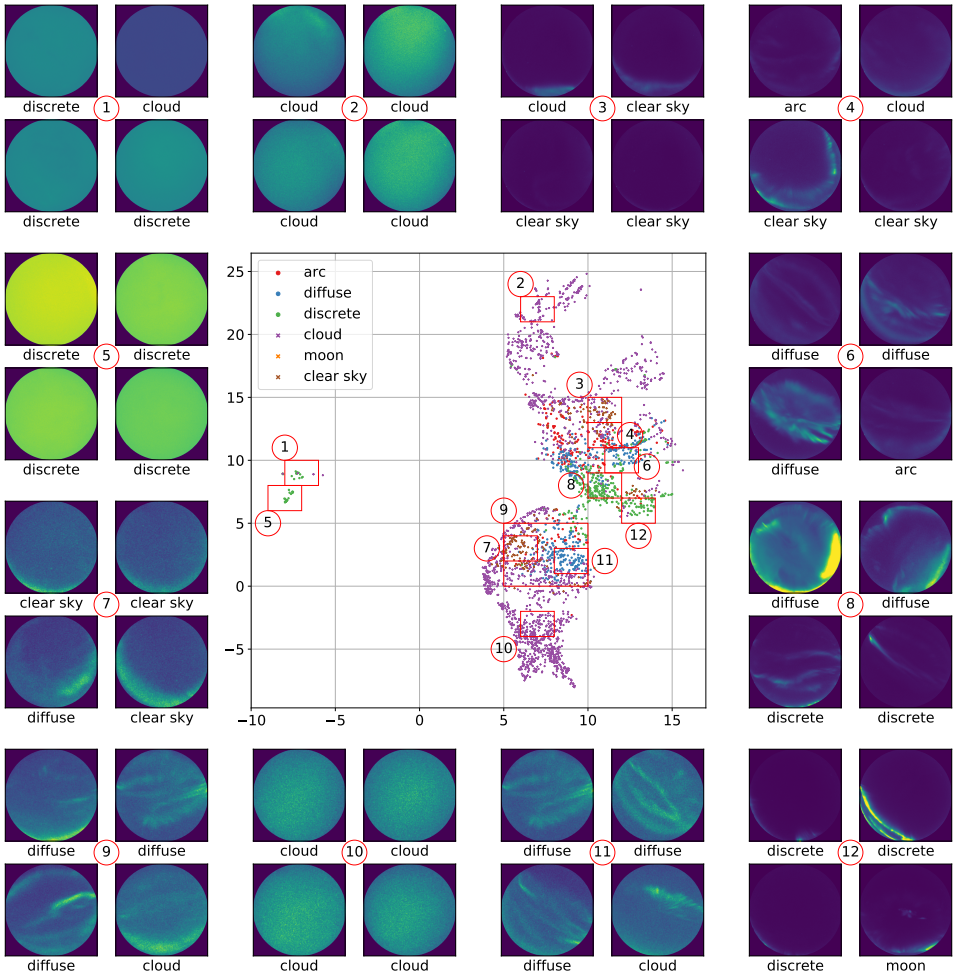


Figure 5: Features embedded into a 2D Space using the UMAP algorithm. Some areas of interest are highlighted by red boxes with 4 random images from within each area shown around the mapping. The numbers by the figures on the outside correspond to the numbers of areas marked in the 2D space.

ping. Each of the groups consists of 4 randomly selected images with the predicted class given below the image and the number assigned to the group in the middle between the images.

The first interesting cluster (areas 1 and 5) is located around (-7, 7) in the coordinate system of the mapping. Most of the images are predicted to be discrete aurora and some to be cloudy images. Upon inspection however we see that all of the images show the dome of the imager covered in snow. All of the images have been clearly misclassified. We can however also see that there are two subclusters, the bottom one (area 5) containing lighter images, the top one (area 1) darker images.

The two clusters around (7, -7) (area 10) and (7, 22) (area 2) contain images which have been classified as cloudy. We can see important optical differences between the two areas: Area 2 contains images where sometimes fuzzy aurora are visible behind the clouds and the cloud cover appears to be quite thin. Area 10 contains darker and more homogeneous images. Although light is sometimes visible through these clouds, they appear to be thicker

than on the other images. Measurements taken with the ceilometer confirm that the cloud base height for clouds in area 2 is between 350 m to 750 m, whereas the cloud base heights for the clouds in area 10 are between 600 m to 1800 m or designated as "cloud free" in about 20% of the images. Although this does not give information about the thickness of the clouds, the difference in altitude could also explain the optical differences in the images.
Those images in areas 2 and 10 are correctly predicted by the classifier to belong to the same class, but according to UMAP they are feature wise different enough to be placed far apart from each other, with clusters of images showing aurora placed in between them.
Two such clusters showing aurora and placed between areas 2 and 10 are the two clusters around (13, 6) (area 12) and (11, 8) (area 8) and show images mostly classified as discrete aurora. For area 12 we can see that the bottom right image has been labelled to show the moon. All of the images in general show light sources near the bottom of the image, but the aurora is usually barely visible. We have already shown above that the classifier has problems discerning images with light-sources near the field of view of the imager, which explains the misclassification of at least one of the images.
Area 8 also shows images that belong in the diffuse class and have been labelled as such.
Around (9, 3) (area 11) we see a cluster of diffuse aurora. It is placed close the one of the previously discussed clusters of cloudy images. The images clearly contain aurora, but are still very similar to the cloudy images.
Another cluster of mostly patchy aurora is placed around (12, 10) (area 6). Those images have a much darker and more homogeneous background but aurora is again clearly visible. The classification of these images would be ambiguous for humans and the classifier also occasionally classifies images as "discrete" or "cloudy" in this cluster.
Next to it is area 4 around (11, 12), where we see a cluster of mostly arcs and non-auroral classes. Because arcs are defined to be thin bands spanning most of the sky, they are optically similar to images showing a clear night sky. The classifier again had problems and was not consistent in assigning the correct labels to images.
Around (11, 14) (area 3) there is a cluster of almost completely dark images. They are located next to the previously analyzed cluster but show no features. Most of these images are predicted to be clear sky images or cloudy. As explained before, there are no completely black images of the night sky available for training. The images showing clouds also always show a uniform, thick cloud cover, which might be why the classifier labels a uniformly black image as cloudy.
At last we have images around (6, 3) (area 7). These are the images which show auroral features near the bottom edge of the image and are labelled as showing no aurora. Many of these images will not be of scientific use, because the aurora are too fuzzy, too weak or make up to little of the image. This cluster contains 16238 of the 550286 images shown in this mapping ($\approx 3.0\%$) which is comparable to the error between aurora classes and the clear sky class ($\approx 2.3\%$) obtained from the testing data of the classifier. The cluster is also bordered by correctly classified images on its right and to its bottom. This means that the amount of images where this problem shows is likely confined to this cluster. While the error is higher than that of the testing data of the oath dataset, because all of the images have not been seen before, we estimate images like this make up less than 5% of the total images, accounting for images that are located somewhere else than this cluster.

4.4 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 modelling 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 magnethosphere 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.

4.4.1 Predicting Cloud Coverage

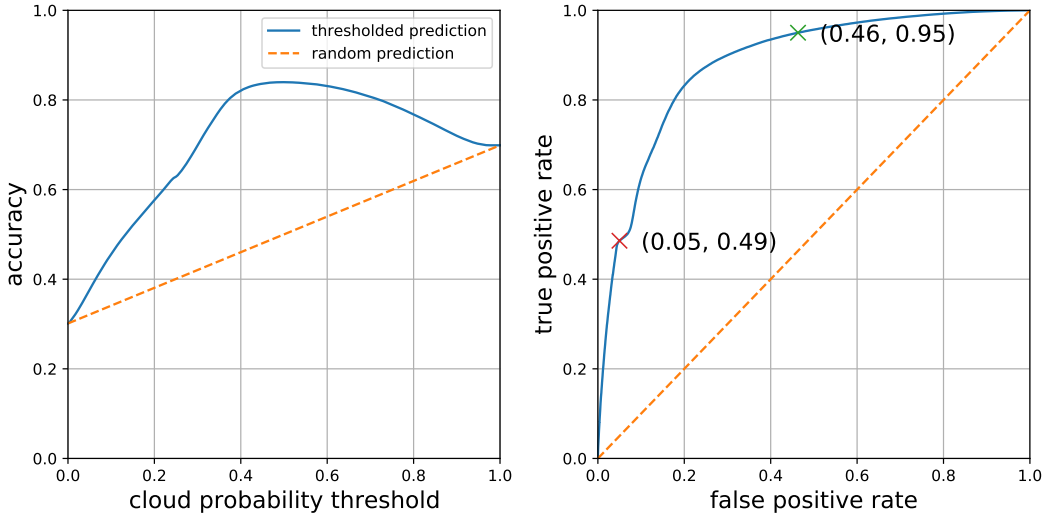


Figure 6: Assessment of the prediction quality for "cloudy" versus "non-cloudy". Panel (a) shows the accuracy as a function of the threshold and Panel (b) shows the ROC curve. In both plots, the continuous blue line shows our thresholded prediction and the orange dotted line corresponds to a random classifier. 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 Panel (b) correspond to a "cloud removal" (green) and "cloud extraction" (red) application scenario.

It is important to note that the ceilometer only measures the Cloud Base Height directly above it, therefore the prediction and measurement might disagree slightly, but it is still good for a general evaluation. Here, every image is assigned the label "cloudy" if the

value returned by the ceilometer is below 2000 m or "not cloudy" otherwise. Clouds at a medium altitude are not considered to make the image cloudy because they do not obstruct enough of the field of view of the classifier.
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 previously unknown to the classifier.
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, i.e. 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 6a. 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 84.0% is achieved at a threshold of 49.6%. Increasing the threshold further, the limit goes towards 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 6a. 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.
In Figure 6b 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 0.88. 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 the curve 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%. In this case, 49% 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. This however leads to falsely labelling 46% 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 labelled test cases. Retraining the classifier on a pure "cloudy" vs "not cloudy" basis could also further improve this technique.

4.4.2 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. 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 2000 m. This gives 178783 images for analysis out of originally 550286 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.

Figure 7 shows the predictions of the perturbations in Earth's local magnetic field based on the extracted features. We have selected 3 events based on their effects on the magnetic field. The first is shown in Figure 7a, which shows an event with weak perturbations in the magnetic field. Figure 7b shows an event with medium sized perturbations and 7c the event where the strongest perturbations have been observed. In all figures the blue connected line shows all data available for the season, whereas the orange dots mark the values used for testing. The green dots then represent the values belonging to these testing data predicted by a linear ridge model trained on the extracted features. The red line shows a running average (n=50) through the predicted values and the purple line a running average through the differences between the predicted and testing data.

Observing the single, weak event on 2010-12-06 (Figure 7a), we see that although the predictive capability per image is low, the running average is able to follow the trend measured by the magnetometer well. An exception to this is the second half of the peak around 10:00UTC, where the prediction already drops to the value observed after the peak. Manually analyzing the images for this point in time yielded no answer as to why this could have happened.

A stronger event on 2011-01-08 (Figure 7b) shows how well the model is able to follow even larger peaks such as the one at around 21:00UTC and keeping the same trend as the testing data before and after the event.

For the last event on 2011-02-05 (Figure 7c) we see much worse predictive capability of the model. Single data points are trying to fit onto the large peaks and we see that the running average is trying to follow the peaks, but only the last peak around 21:00UTC can be somewhat matched. This comes to no surprise because this is the strongest event observed during the whole season. Out of the approximately 200000 datapoints used for training, only a few hundred belong to this event. The model has therefore not seen enough data like this to be able to accurately predict it.

In Figure 7d we show a scatter plot of the predicted data vs the testing data. In blue we show points taken on their own, in orange we show the time-averaged data as explained previously.

For both cases, the points' center of mass lies in the origin and 96% of the points are within a 100 nT radius around the origin. These points correspond to events as displayed in Figure 7a and Figure 7b except for the large peak in the beginning.

Ideally every point would fall on the black line that is drawn where $B_{test} = B_{prediction}$. For points closer to the origin, the smoothed predictions are closer to the line, whereas the single predictions show larger errors. Towards larger negative test values, the predicted values underestimate the testing values in most cases.

In Table 3 we give the errors and coefficients of determination obtained for each event and the whole season. The data for the whole season as well as for the weak and medium events produces lower errors when the running average of the data is taken instead of every point of data on its own. Only for the largest event this method performs worse. A likely reason is that the width of the running average of n=50 is larger than the width of the peaks that are to be predicted. Strong but narrow peaks can not accurately be smoothed onto, even if the single values perfectly matched the data.

Over the whole season the coefficient of determination is $R^2 = 0.57$ for the smoothed data.

Table 3: Mean squared errors and absolute errors for single values and the running average of the prediction for the whole season and 3 selected events.

	figure	Event length [# of test-datapoints]	Mean squared error [nT]		Mean absolute error [nT]		Coefficient of Determination (R^2)	
			single value	running average	single value	running average	single value	running average
all data		35 706	1208.4	975.0	20.3	15.0	0.47	0.57
weak event	7a	1000	348.1	119.4	13.8	8.0	−1.61	0.10
medium event	7b	1290	942.2	788.6	23.5	19.4	0.16	0.29
large event	7c	320	25 982.3	29 938.2	109.8	113.3	−0.01	−0.17

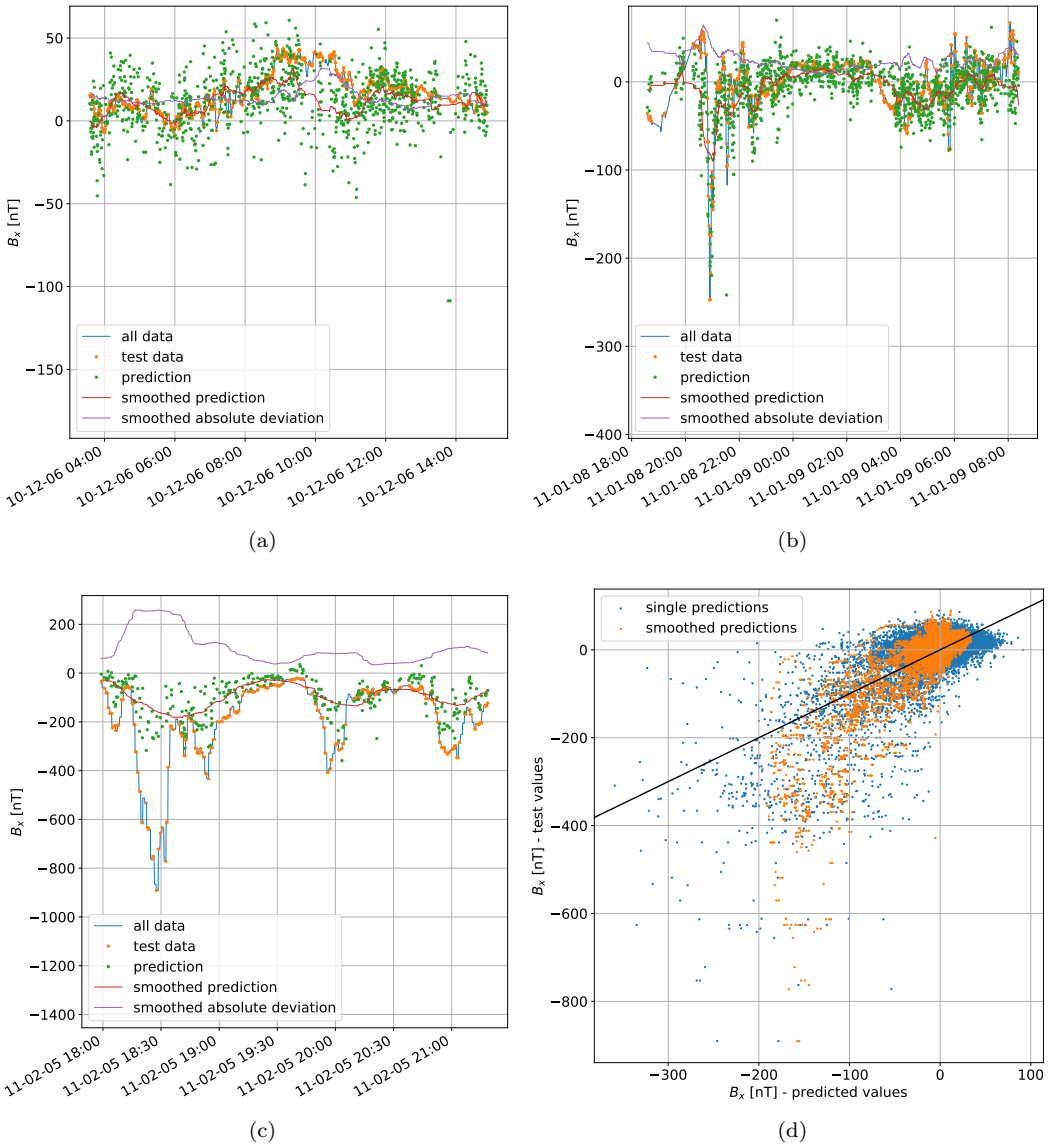


Figure 7: Predictions of the magnetic field component in the X direction for the 2010 / 2011 season, based on features extracted by the neural network. Panel (a) shows a small event during the season, Panel (b) an event of medium strength and Panel (c) the strongest perturbation event during the season. Note that the scale of the y-axis differs. Panel (d) shows a scatter plot of the predicted vs the testing data for single data points and their smoothed values.

For the selected weak and medium events, this value is lower each, but the coefficient obtained for the smoothed data is in each case better. The large event performs bad in general. Here the smoothing had a negative effect on the coefficient, likely for the same reasons why the errors also increased for the smoothed data.

Overall, we find that that our model cannot predict the perturbations on a single-image-basis but performs well when modelling the trend of the perturbations based on single-image-predictions. The model performs best for small perturbations $< 100\,\mathrm{nT}$ because 96%

of the data lie within this range. Larger perturbations are indicated, but cannot be matched due to the nature of the model and unavailability of training data.

5 Conclusion and Outlook

Based on the Oslo Auroral THEMIS (OATH) dataset, containing 5824 labelled 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 support vector machine, where the best hyperparameters have been evaluated using tenfold cross validation. We achieve $89.58 \pm 0.90\%$ accuracy in the six-class classification, and $97.81 \pm 0.41\%$ when aggregating images into two classes "aurora" and "no aurora" after classification.
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 unlabelled images taken in Ny Ålesund between Nov 2010 and Feb 2011. To test the classifier, we analyse 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 labelled ambiguously by the classifier, most likely because of the same problem in the training data as for the testing data. Improving the training data by introducing more labelled images from different sources and allowing multi labelled images (e.g "arcs & clouds") could lead to an improvement in prediction.
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 84% 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 the magnetic field values, we find that the predictive capability of a single image is not enough to accurately predict the magnetic field values, however a rolling average 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, modelling 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.
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. The classifier is also made publicly available under the Attribution 4.0 International (CC BY 4.0) license giving anyone the possibility to classify images with just a few lines of code. GPU based feature extraction followed by classification could be applied to millions of images in a week, showing that this is a viable approach for our datasets and others like the THEMIS images.

Acknowledgments

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/10.11582/2021.00057` and encourage anyone to use our classifier. 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).

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[3]. 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 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 the gridsearch performed to find the best hyperparameters. In order to emphasize the plateau for which the highest performance is achieved, we have decided to cut off accuracies below 85%. The jagged edges in the top area of the figure are artefacts due to interpolation and the cut-off, not reflecting the actual data. The classifier performs best for $\gamma = 0.001$ and $C = 10$. For higher values of C and constant γ almost the same accuracy on the data can be achieved, but increasing C would likely lead to overfitting on the training data and decreased classification performance on unseen data.

References

Akasofu, S.-I. (1964). The development of the auroral substorm. *Planetary and Space Science*, *12*(4), 273-282. Retrieved from `https://www.sciencedirect.com/science/article/pii/0032063364901515` doi: https://doi.org/10.1016/0032-0633(64)90151-5

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

[3] `https://pytorch.org/vision/stable/index.html`

Table A1: Overview of all tested pretrained neural networks. 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 normalisation values for the provided image, "diff" and "mem" the time in seconds and used memory in MB to extract all features, "acc_class2", "dev_class2", "acc_class6", "dev_class6" the accuracies and standard deviations for 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.

	model	key	num_classes	size	mean	std	diff	mem	acc_class2	dev_class2	acc_class6	dev_class6
0	fbresnet152_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	68.2068	2175.0000	0.9513	0.0039	0.8015	0.0067
1	bninception_cadene	imagenet	1000	224	[104. 117. 128.]	[1. 1. 1.]	45.9797	3369.0000	0.8815	0.0078	0.6857	0.0066
2	resnext101_32x4d_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	65.0266	3369.0000	0.9580	0.0041	0.8042	0.0048
3	resnext101_64x4d_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	83.1215	4153.0000	0.9484	0.0018	0.8066	0.0093
4	inceptionv4_cadene	imagenet	1000	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	73.1479	4153.0000	0.9504	0.0037	0.7969	0.0071
5	inceptionv4_cadene	imagenet+background	1001	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	71.0594	4153.0000	0.9483	0.0021	0.7973	0.0070
6	inceptionresnetv2_cadene	imagenet	1000	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	77.2813	4189.0000	0.9568	0.0014	0.8185	0.0032
7	inceptionresnetv2_cadene	imagenet+background	1001	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	78.1985	4189.0000	0.9550	0.0028	0.8215	0.0082
8	alexnet_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	40.4227	4189.0000	0.9343	0.0047	0.7815	0.0066
9	densenet121_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	48.8951	4189.0000	0.9573	0.0058	0.8247	0.0092
10	densenet169_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	51.6538	4193.0000	0.9582	0.0040	0.8063	0.0059
11	densenet201_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.1744	4257.0000	0.9582	0.0044	0.8189	0.0072
12	densenet161_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	64.7054	4313.0000	0.9583	0.0028	0.8284	0.0050
13	resnet18_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	43.8311	4313.0000	0.9492	0.0044	0.7922	0.0082
14	resnet34_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	46.2278	4313.0000	0.9462	0.0039	0.7936	0.0075
15	resnet50_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	50.5016	4321.0000	0.9524	0.0029	0.8096	0.0020
16	resnet101_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	59.4153	4373.0000	0.9573	0.0025	0.8088	0.0077
17	resnet152_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	63.5194	4455.0000	0.9571	0.0037	0.8133	0.0070
18	inceptionv3_cadene	imagenet	1000	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	59.1663	4493.0000	0.9548	0.0023	0.8219	0.0027
19	squeezenet1_0_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	44.2880	4495.0000	0.9554	0.0052	0.8198	0.0048
20	squeezenet1_1_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	42.5836	4497.0000	0.9517	0.0023	0.8196	0.0041
21	vgg11_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	51.5564	5281.0000	0.9496	0.0024	0.7828	0.0063
22	vgg11_bn_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	49.1940	6065.0000	0.9442	0.0020	0.7981	0.0051
23	vgg13_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	52.7669	6065.0000	0.9446	0.0040	0.7897	0.0037
24	vgg13_bn_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	54.1080	6849.0000	0.9448	0.0019	0.7829	0.0093
25	vgg16_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	56.4526	6849.0000	0.9342	0.0045	0.7756	0.0096
26	vgg16_bn_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.4583	6849.0000	0.9449	0.0033	0.7785	0.0070
27	vgg19_bn_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	58.5755	6849.0000	0.9445	0.0038	0.7731	0.0049
28	vgg19_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	56.6498	7633.0000	0.9386	0.0023	0.7710	0.0042
29	nasnetamobile_cadene	imagenet	1000	224	[0.5 0.5 0.5]	[0.5 0.5 0.5]	50.6985	7633.0000	0.9512	0.0022	0.8076	0.0112
30	nasnetalarge_cadene	imagenet	1000	331	[0.5 0.5 0.5]	[0.5 0.5 0.5]	159.2706	11131.0000	0.9517	0.0023	0.8163	0.0095
31	nasnetalarge_cadene	imagenet+background	1001	331	[0.5 0.5 0.5]	[0.5 0.5 0.5]	161.6795	11131.0000	0.9573	0.0027	0.8254	0.0036
32	dpn68_cadene	imagenet	1000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	52.6899	11131.0000	0.9442	0.0039	0.7962	0.0071
33	dpn68b_cadene	imagenet+5k	1000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	52.3839	11131.0000	0.9542	0.0037	0.8159	0.0061
34	dpn92_cadene	imagenet+5k	1000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	62.0724	11131.0000	0.9588	0.0031	0.8068	0.0040
35	dpn98_cadene	imagenet	1000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	77.0900	11131.0000	0.9471	0.0069	0.7915	0.0053
36	dpn131_cadene	imagenet	1000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	88.3290	11131.0000	0.9460	0.0042	0.8003	0.0062
37	dpn107_cadene	imagenet+5k	1000	224	[0.4863 0.4588 0.4078]	[0.2348 0.2348 0.2348]	93.2658	11131.0000	0.9494	0.0018	0.7943	0.0040
38	xception_cadene	imagenet	1000	299	[0.5 0.5 0.5]	[0.5 0.5 0.5]	71.9872	11131.0000	0.9525	0.0072	0.8184	0.0071
39	senet154_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	101.3283	4105.0000	0.9576	0.0045	0.8316	0.0057
40	se_resnet50_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	50.6819	4105.0000	0.9606	0.0029	0.8353	0.0059
41	se_resnet101_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	57.3070	4123.0000	0.9651	0.0045	0.8282	0.0088
42	se_resnet152_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	67.7151	4159.0000	0.9598	0.0031	0.8366	0.0049
43	se_resnext50_32x4d_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.1590	4159.0000	0.9562	0.0034	0.8228	0.0044
44	se_resnext101_32x4d_cadene	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	66.5890	4159.0000	0.9629	0.0031	0.8247	0.0064
45	cafferesnet101_cadene	imagenet	1000	224	[102.9801 115.9465 122.7717]	[1. 1. 1.]	55.7170	4159.0000	0.8600	0.0057	0.6546	0.0080
46	pnasnet5large_cadene	imagenet	1000	331	[0.5 0.5 0.5]	[0.5 0.5 0.5]	158.9650	8331.0000	0.9434	0.0066	0.7961	0.0079
47	pnasnet5large_cadene	imagenet+background	1001	331	[0.5 0.5 0.5]	[0.5 0.5 0.5]	158.2496	8331.0000	0.9465	0.0049	0.7955	0.0080
48	polynet_cadene	imagenet	1000	331	[0.485 0.456 0.406]	[0.229 0.224 0.225]	136.7295	9367.0000	0.9551	0.0065	0.8188	0.0096
49	alexnet_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	40.2826	9367.0000	0.9368	0.0034	0.7863	0.0078
50	vgg11_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	51.1166	9367.0000	0.9477	0.0016	0.7869	0.0099
51	vgg11_bn_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	48.9552	9367.0000	0.9476	0.0049	0.7989	0.0044
52	vgg13_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	52.3303	9367.0000	0.9442	0.0032	0.7858	0.0064
53	vgg13_bn_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	54.1776	9367.0000	0.9475	0.0052	0.7912	0.0128
54	vgg16_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.1364	9367.0000	0.9291	0.0069	0.7782	0.0045
55	vgg16_bn_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	59.0248	9367.0000	0.9452	0.0041	0.7804	0.0045
56	vgg19_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	56.4227	9367.0000	0.9383	0.0021	0.7749	0.0035
57	vgg19_bn_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	57.9345	9367.0000	0.9408	0.0057	0.7742	0.0080
58	resnet18_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	42.8714	9367.0000	0.9472	0.0036	0.7927	0.0073
59	resnet34_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	45.7796	9367.0000	0.9477	0.0030	0.7923	0.0042
60	resnet50_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	52.6868	9367.0000	0.9524	0.0053	0.8061	0.0051
61	resnet101_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.1196	9367.0000	0.9543	0.0026	0.8160	0.0128
62	resnet152_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	62.0644	9413.0000	0.9578	0.0045	0.8049	0.0049
63	squeezenet1_0_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	43.7105	9417.0000	0.9562	0.0033	0.8182	0.0041
64	squeezenet1_1_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	41.6251	9419.0000	0.9483	0.0021	0.8124	0.0075
65	densenet121_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	53.7737	9451.0000	0.9571	0.0013	0.8243	0.0086
66	densenet169_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	53.0848	9503.0000	0.9578	0.0024	0.8112	0.0085
67	densenet161_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	60.5094	9559.0000	0.9607	0.0050	0.8247	0.0121
68	densenet201_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	54.9551	9617.0000	0.9614	0.0033	0.8169	0.0095
69	inception_v3_torchvision	imagenet	1000	299	[0.485 0.456 0.406]	[0.229 0.224 0.225]	60.5731	9641.0000	0.9515	0.0030	0.8066	0.0108
70	googlenet_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	43.8201	9653.0000	0.9508	0.0040	0.8141	0.0048
71	shufflenet_v2_x0_5_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	39.8624	9653.0000	0.9539	0.0048	0.8190	0.0048
72	shufflenet_v2_x1_0_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	40.6544	9655.0000	0.9551	0.0029	0.8306	0.0044
73	mobilenet_v2_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	44.5089	9663.0000	0.9523	0.0036	0.8065	0.0081
74	resnext50_32x4d_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	57.5930	9671.0000	0.9526	0.0027	0.8072	0.0052
75	resnext101_32x8d_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	76.8500	9683.0000	0.9602	0.0042	0.8195	0.0051
76	wide_resnet50_2_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	55.4168	9691.0000	0.9534	0.0017	0.8096	0.0053
77	wide_resnet101_2_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	65.0305	9701.0000	0.9483	0.0021	0.8061	0.0086
78	mnasnet0_5_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	42.2927	9703.0000	0.9555	0.0047	0.8230	0.0079
79	mnasnet1_0_torchvision	imagenet	1000	224	[0.485 0.456 0.406]	[0.229 0.224 0.225]	47.8704	9713.0000	0.9668	0.0038	0.8295	0.0079

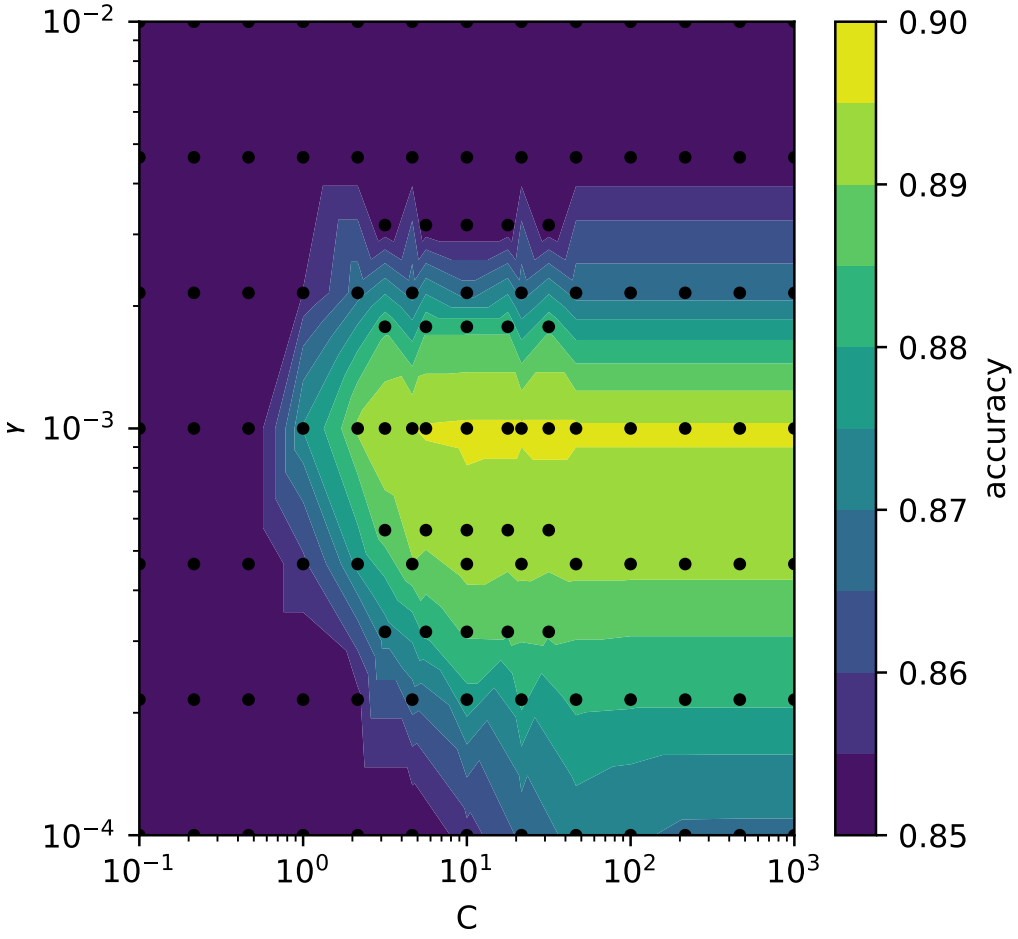


Figure B1: Performance for hyperparameters C and γ evaluated using 10 fold cross validation

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. doi: 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, August). Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, *42*(8), 2011–2023. Retrieved from https://doi.org/10.1109/TPAMI.2019.2913372 doi: 10.1109/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)* (p. 2261-2269). doi: 10.1109/CVPR.2017.243

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017, May). ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM*, *60*(6), 84–90. Retrieved from https://doi.org/10.1145/3065386 doi: 10.1145/3065386

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. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020JA027808 (e2020JA027808 10.1029/2020JA027808) doi: 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). Cham: Springer International Publishing.

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. Retrieved from https://essd.copernicus.org/articles/10/1451/2018/ doi: 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. Retrieved from 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) doi: 10.1594/PANGAEA.880300

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, *3*(29), 861. Retrieved from https://doi.org/10.21105/joss.00861 doi: 10.21105/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. Retrieved from https://gi.copernicus.org/articles/9/267/2020/ doi: 10.5194/gi-9-267-2020

McPherron, R. L., Aubry, M. P., Russell, C. T., & Coleman Jr., P. J. (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. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JA078i016p03068 doi: https://doi.org/10.1029/JA078i016p03068

Mende, S. B., Harris, S. E., Frey, H. U., Angelopoulos, V., Russell, C. T., Donovan, E., ... Peticolas, L. M. (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). New York, NY: Springer New York. Retrieved from https://doi.org/10.1007/978-0-387-89820-9_16 doi: 10.1007/978-0-387-89820-9_16

Murphy, K. R., Mann, I. R., Rae, I. J., Waters, C. L., Frey, H. U., Kale, A., ... Korth, H. (2013). The detailed spatial structure of field-aligned currents comprising the substorm current wedge. *Journal of Geophysical Research: Space Physics*, *118*(12),

7714-7727. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2013JA018979 doi: https://doi.org/10.1002/2013JA018979

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (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., ... Milling, D. K. (2017). Using ultra-low frequency waves and their characteristics to diagnose key physics of substorm onset. *Geoscience letters*, *4*(1), 1–11.

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

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... others (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), 211–252.

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.* (p. 289-295). doi: 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. Retrieved from https://www.sciencedirect.com/science/article/pii/S0273117701004926 doi: 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* (p. 1063-1066). doi: 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. Retrieved from https://angeo.copernicus.org/articles/22/1103/2004/ doi: 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*, 224–228.

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

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, Feb.). 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)* (p. 2818-2826). doi: 10.1109/CVPR.2016.308

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). MnasNet: Platform-Aware Neural Architecture Search for Mobile. In *2019 IEEE conference on computer vision and pattern recognition (CVPR)* (p. 2815-2823). doi:

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: Space Physics*, *114*(A5). Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JA013682 doi: 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. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JA026119 doi: 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. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925231217318180 doi: 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. Retrieved from https://www.sciencedirect.com/science/article/pii/S0893608019301108 doi: https://doi.org/10.1016/j.neunet.2019.04.012