

Earth and Space Science



**TECHNICAL
REPORTS: METHODS**
10.1029/2021EA002007

Key Points:

- Atmospherics can be effectively detected on frequency-time spectrograms by a deep learning approach
- Our method provides automatic and reliable extraction of desired atmospherics' details suitable for further statistical analysis
- The developed method is generally applicable as a methodology for similar applications in the future

Correspondence to:

Š. Mackovjak,
mackovjak@saske.sk

Citation:

Maslej-Krešňáková, V., Kundrát, A., Mackovjak, Š., Butka, P., Jaščur, S., Kolmašová, I., & Santolík, O. (2021). Automatic detection of atmospherics and tweek atmospherics in radio spectrograms based on a deep learning approach. *Earth and Space Science*, 8, e2021EA002007. <https://doi.org/10.1029/2021EA002007>

Received 8 SEP 2021
Accepted 27 OCT 2021

Author Contributions:

Conceptualization: Viera Maslej-Krešňáková, Šimon Mackovjak, Peter Butka, Ivana Kolmašová, Ondřej Santolík

Data curation: Viera Maslej-Krešňáková, Ivana Kolmašová, Ondřej Santolík








Formal analysis: Viera Maslej-Krešňáková, Adrián Kundrát, Samuel Jaščur

Funding acquisition: Šimon Mackovjak, Peter Butka

© 2021 The Authors. Earth and Space Science published by Wiley Periodicals LLC on behalf of American Geophysical Union.

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs License](#), which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

Automatic Detection of Atmospherics and Tweek Atmospherics in Radio Spectrograms Based on a Deep Learning Approach

Viera Maslej-Krešňáková¹ , Adrián Kundrát¹ , Šimon Mackovjak² , Peter Butka¹ , Samuel Jaščur¹ , Ivana Kolmašová^{3,4} , and Ondřej Santolík^{3,4} 

¹Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Košice, Slovakia, ²Department of Space Physics, Institute of Experimental Physics, Slovak Academy of Sciences, Košice, Slovakia, ³Department of Space Physics, Institute of Atmospheric Physics of the Czech Academy of Sciences, Prague, Czechia, ⁴Faculty of Mathematics and Physics, Charles University, Prague, Czechia

Abstract Lightning strikes can be routinely observed by the radio receivers that measure electromagnetic signals in the very low frequency range. The acquired pulses called atmospherics provide valuable details about source lightning discharges and also about the Earth-ionosphere waveguide where they can propagate thousands of kilometers. The automatic acquisition of the events requires also automatic methods for extraction of atmospherics' details to confirm the observed trends with statistical significance. For this purpose, we have developed a method based on a deep learning approach to automatically obtain the type of atmospherics, their exact time, and the frequency range from the frequency-time spectrograms. The method that employs convolutional neural networks is designed to be adaptable to scientific needs and provide reliable results according to the requirements on the sensitivity to events, computation performance, and precision of extracted details. The efficiency and specific steps of our method are demonstrated for data recorded by the ELMAVAN-G instrument. The comprehensive description of the method allows its usage for regular characterization of the ionospheric D-layer or for other similar applications in the future.

1. Introduction

Lightning discharges emit electromagnetic pulses in a broad range of frequencies with a peak of radiated power in the very low frequency (VLF) range. The lightning generated VLF signals called atmospherics (or sferics for short) propagate over thousands of kilometers, being reflected from the boundaries of the Earth-ionosphere waveguide, which is formed by the conductive ground or ocean on one side and by the bottom of the conductive ionosphere on the other side. These signals are routinely recorded and analyzed for various purposes. Obtained lightning flash rates and flash densities can be used to derive thundercloud properties (Deierling & Petersen, 2008; Huffines & Orville, 1999; Williams et al., 2000), to predict the severity of tornadic storms (Chronis et al., 2015) and hurricanes (Price et al., 2009), or to estimate amounts of lightning-produced NO_x (Schumann & Huntrieser, 2009).

The characteristics of electromagnetic signals emitted by lightning and propagating in the Earth-ionosphere waveguide reflect the properties of the waveguide. Its upper boundary—the D-region of the ionosphere at altitudes between 50 and 90 km—is influenced from above by the Sun. The altitude is reflected by pronounced day/night differences of the properties of the ionospheric D-region or by the effects of solar flares (Grubor et al., 2008). The D-region can also be modified from below by powerful meteorological (Haldoupis et al., 2013; Inan et al., 2010; Kolmašová et al., 2021; Mezentsev et al., 2018) or seismogenic sources (Píša et al., 2013; Pulinets & Boyarchuk, 2005). Investigation of this region of the ionosphere is difficult, as it is not accessible to orbiting spacecraft, and cannot be monitored easily by ionosondes or scatter radars (Thomson & McRae, 2009). The available balloon or rocket measurements provide only temporally and spatially limited information about the properties of the ionospheric D-layer. As there are about 50 lightning flashes which hit the ground every second (Christian et al., 2003), analysis of the propagation of lightning related signals can serve as a tool for the investigation of the bottom of the ionosphere (Cummer et al., 1998; Han & Cummer, 2010; Said et al., 2010) also providing useful information for propagation models of VLF radio waves (Jacobson et al., 2021; Rapoport et al., 2020).

Investigation: Viera Maslej-Krešňáková, Adrián Kunderát, Samuel Jaščur

Methodology: Viera Maslej-Krešňáková, Šimon Mackovjak, Peter Butka

Project Administration: Šimon Mackovjak

Resources: Šimon Mackovjak, Peter Butka

Software: Viera Maslej-Krešňáková, Adrián Kunderát, Šimon Mackovjak, Samuel Jaščur

Supervision: Šimon Mackovjak, Peter Butka, Ivana Kolmašová, Ondřej Santolík

Validation: Šimon Mackovjak, Samuel Jaščur, Ivana Kolmašová, Ondřej Santolík

Visualization: Viera Maslej-Krešňáková, Adrián Kunderát, Samuel Jaščur

Writing – original draft: Viera Maslej-Krešňáková, Adrián Kunderát, Šimon Mackovjak

Writing – review & editing: Šimon Mackovjak, Peter Butka, Ivana Kolmašová, Ondřej Santolík

According to the waveguide theory (Budden, 1961), a lightning generated signal excites different transmission waveguide modes. A zero-order mode propagates in the waveguide at all frequencies. Higher-order modes cannot travel below their critical frequencies. Night-side sferics usually show typical frequency dispersion signatures which produce clicking sounds in the loudspeakers of the receivers and are therefore known as “tweeks” (Helliwell, 1965). Ionospheric reflection heights, propagation distances and relevant electron densities can be derived from the frequency-time characteristics of tweeks. This method has therefore been used in several case studies to investigate the morphology of the nighttime (Kumar et al., 2008; Maurya et al., 2012; Ohya et al., 2003; Tan, 2016) and exceptionally also the daytime (Ohya et al., 2015; Santolík & Kolmašová, 2017) D-region of the ionosphere.

A large amount of data is needed for statistical studies to confirm the observed trends. Machine learning techniques can significantly simplify the detection and analysis of the sferics and tweeks or whistlers (Harid et al., 2021) recorded in the form of frequency-time spectrograms. The research presented below focuses on an automatic method for detection and localization of specific signals within frequency-time spectrograms based on convolutional neural networks (LeCun et al., 2015). We have employed a very effective approach of computer vision with a focus on bounding box regression. To select the most effective approach, we studied Konan et al. (2020), who provide the implementation of three different methods for detection and localization of whistler radio waves. The first method called CCSW used the whistler's dispersion approximation developed by Bernard (1973) and applied for whistler wave detection in Lichtenberger et al. (2008). The next two methods—SDCNN (Chang et al., 2019) and YOLOv3 (Redmon et al., 2016) are based on convolutional neural networks and proved their effectiveness for the detection of events. In O'Shea et al. (2017), the authors describe the application of the smallest YOLO architecture for the radio signal detection. This research introduced an approach to train a highly effective radio signal detection method that achieves higher levels of contextual understanding than traditional energy thresholding-based signal detection schemes. The universal abilities of deep learning architecture can be easily seen also in Fanioudakis and Potamitis (2017), which focuses on reliable detection and segmentation of bird vocalizations as recorded in the open field. Input data are in the form of audio spectrogram of bird vocalizations. The researchers in this area implemented automatized monitoring of multiple bird subspecies using two approaches. In the first case, they developed a combination of DenseNets (Huang et al., 2018) and YOLOv2 architecture. In the second approach, they applied U-net architecture (Ronneberger et al., 2015). The use of both techniques has proven to be very effective. The existing works provided a technical view of the methods and their practical aspects, for example, how to process inputs, annotations, and the basic setup of the methods. According to the described studies, YOLO architecture provides a more robust solution and is suitable for our needs.

The paper is organized as follows: Section 2 provides a description of our frequency-time spectrograms, events, and the process of annotation; Section 3 describes the developed method and consists of two parts. The first part presents automatic event detection on spectrograms based on a deep learning approach. The second part describes post-processing—deterministic methods of extraction of the required details from spectrograms. Section 4 presents the performance metrics used in the experiments and discusses the results. The main findings are summarized in Section 5.

2. Data

In this work, we use the frequency-time spectrograms calculated from the 3-component electromagnetic waveforms recorded by the ELMAVAN-G instrument (Santolík & Kolmašová, 2017). The instrument was developed by the Institute of Atmospheric Physics of the Academy of Sciences of the Czech Republic for the Resonance satellite project (Mogilevsky et al., 2012) for the Earth's radiation belts. The instrument consists of a radio receiver coupled with two perpendicular magnetic loop antennas with 12 turns, each with an area of 4 m² and a spherical electric antenna with a diameter of 10 cm, located 2 m above the ground. The ELMAVAN-G instrument is installed on the top of La Grande Montagne near the town of Rustrel in France in an electromagnetically quiet environment of the external site of the Laboratoire Souterrain a Bas Bruit (LSBB). The vertical electric field component and two components of the horizontal magnetic field are sampled at 50 kHz; the duration of each 3-component waveform snapshot is 144 s. The snapshots are stored in the internal memory of the instrument with a cadence of 5 min.

ELMAVAN 3D WAVEFORMS (Measured data sampled at 50 kHz) 49602 packets of 144 samples from 2015-04-09T00:01:15.454.000. Part 47/143
Processed Thu Apr 9 02:08:46 2015 by ELM ver.2012-10-06 from 001_elm20150409_000114__dat00 bin

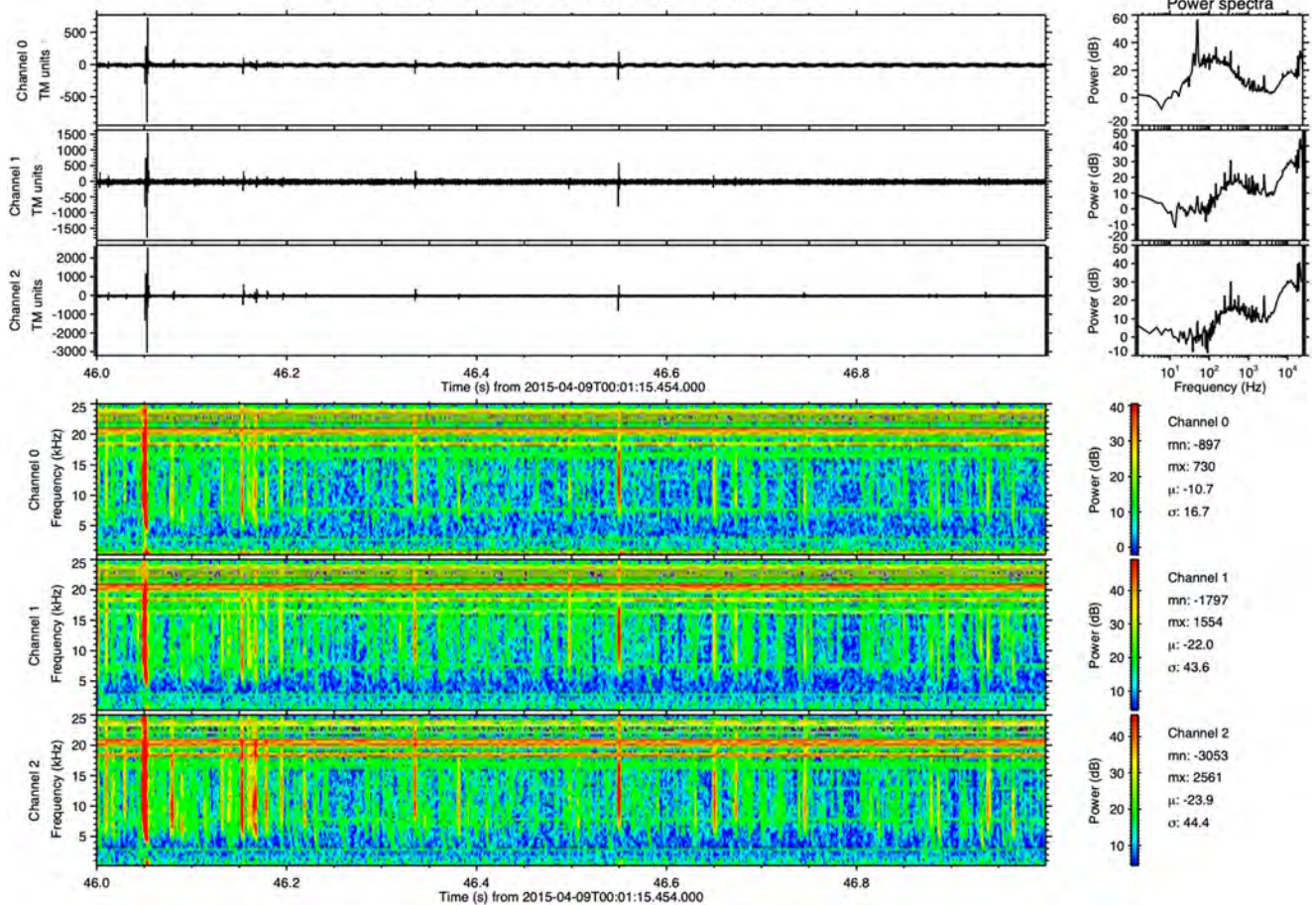


Figure 1. Sample of one page from pdf provided by the initial processing of ELMAVAN-G instrument. Every spectrogram displays one second of the measured interval (144 s) with the strongest emission. In the preprocessing phase, we cropped spectrograms from Channel 2 and used them to train our models. See text for more details on processing of spectrogram pdf file.

The data set for machine learning technique described in Section 3 is composed of images of frequency-time spectrograms generated for each snapshot. Each image shows a 1-s long spectrogram spanning the frequency range from 0 to 25 kHz. The spectrograms are obtained from waveform data using a 256 point FFT with 94% overlapping. Raw digitized data without any pre-processing are used for calculating the spectra. In case of a presence of interferences in the received signal a cleaning procedure can be applied or the color scale of the spectrograms might be adapted. Nine images are generated for each 144-s long snapshot, containing time intervals when the strongest electromagnetic emissions occurred within it. These strong emissions correspond predominantly to broadband lightning sferics and tweeks and usually occupy a substantial part of the whole frequency band. Narrowband electromagnetic signals emitted by powerful VLF transmitters dedicated for communication with submarines are also well recognizable above 16 kHz. For the future statistical scientific analysis all available data will be analyzed. As the yearly amount of data exceeds 15 million spectrogram images, machine learning techniques could provide a helpful tool for automatic detection of lightning events in individual spectrograms.

The data preparation phase consisted of two steps: extracting spectrograms with the necessary information from the stored data and manual labeling of target events within the created crowdsourcing annotation project. Figure 1 displays the example of spectrogram images. We obtained measurement data for the exact second with the strongest recorded emissions by parsing the original image. The most significant events are recorded in Channel 2, so this channel became the basis for our method. Therefore, we cropped spectrograms

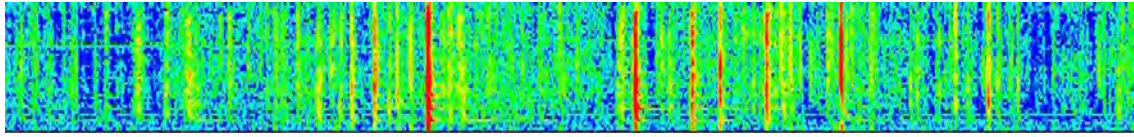


Figure 2. Example of a cropped spectrogram image, which is ready as an input for the training of our neural network model.

and extracted only the selected channel in jpg format. The names of images contain information about the starting time of the spectrogram in the form YYYYMMDD_HHMMSS.

Figure 1 also shows horizontal lines that are on each spectrogram at the position 16–25 kHz. The horizontal lines are signals from low-frequency power transmitters used to communicate with submarines and have nothing to do with the monitored events. Based on this fact, we cropped this part of the spectrogram to prevent possible problems in the neural network's learning process and we decided to use only frequencies lower than 16 kHz from each spectrogram to annotate and train the neural network. At the same time, we kept the scale on the right side of the pictures for further analysis (see Figure 2). We randomly selected 2,500 spectrograms from the four warmest months of the year (July - September) in 2015–2019. The main reason is that storm activity is most intense in Europe during these months. The images prepared in this way became the source for the next step—data annotation.

For supervised learning of neural networks, it is needed to annotate input data. In our case, the annotated data set is a set of spectrograms with labeled events, based on which the model should extract the necessary features for their detection. Because we had a set of spectrogram images without annotations, we used the crowdsourcing approach to obtain them. Brabham (2008) defines crowdsourcing as delegating a problem to a larger mass of unknown people to solve it. We created an annotation project on the Zooniverse platform (zooniverse.org, last access: Sep 1, 2021), where we obtained annotations for training the neural network. The Zooniverse platform focuses on helping researchers who need effective collaboration in their research activities. The main advantage of this platform is that it allows almost anyone to participate in research activities. Also, the platform does not require additional education or hardware accessories for its usage. However, it is essential to provide at least some tutorials or hands-on sessions for people that participate in the specific project. More than 50 students from the Faculty of Electrical Engineering and Informatics of the Technical University in Košice participated in this crowdsourcing project and provided us with a sufficient amount of annotations. In the first phase of the annotation process, it was necessary to indicate whether the event is spheric or a tweek (with a specified number of modes). In the second part, it was annotated whether the signal reaches frequencies below 2 kHz. Figure 3 provides examples of annotated events and information in spectrograms.

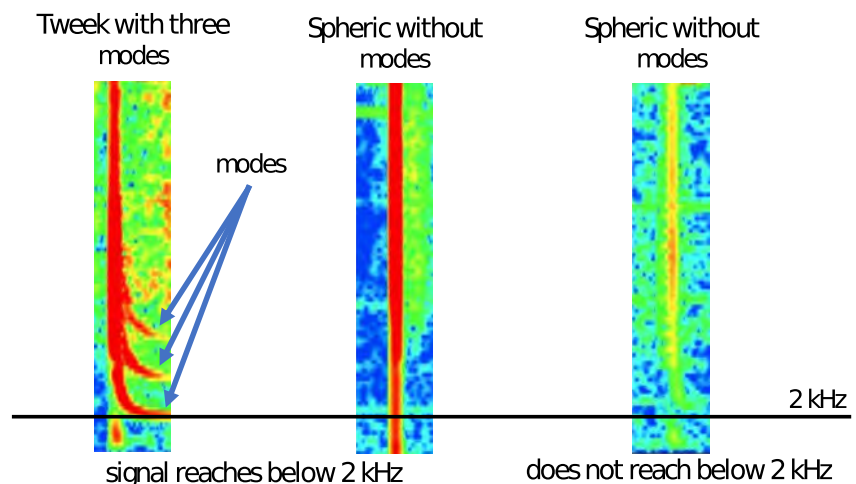


Figure 3. Example of events for the annotation. The first picture represents a tweek that reaches frequency areas below 2 kHz and has three modes. The second and third images provide an example of spherics, where one of them has detected emission below 2 kHz, and another one does not.

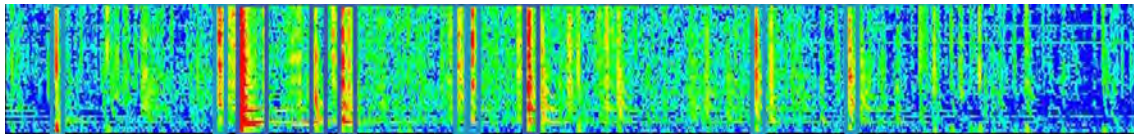


Figure 4. Example of annotations obtained from the Zooniverse project. Dark blue bounding boxes represent tweeks and light blue boxes sferics.

After visual verification, we used 2,300 images in the final data set, with labeled events of 3,200 tweeks and 19,000 sferics. Figure 4 provides an example of the acquired annotations. The Zooniverse platform generates annotation information in the form of JSON files. The values from the JSON file represent output parameters values as class x_center y_center width height, where the first value is the class of event (sferic or tweek), the other two values represent the coordinates for the center of the labeling bounding box, then the width and height of the box. For the training process, we normalized bounding box data by scaling between 0 and 1. The last step of data pre-processing is dividing the final input data into training, validation, and testing subsets. Table 1 provides the number of images and events in each subset.

3. Methods

3.1. Automatic Events Detection

Our main goal is to provide a method for the automatic detection of events in frequency-time spectrograms. We decided to apply methods known from the field of computer vision based on object detection algorithms. Such techniques automatically process the images and provide coordinates of the bounding boxes around the objects and their class in the image. The most successful approaches from recent years have been deep learning methods based on deep neural network architectures (L. Liu et al., 2020). Several multi-purpose architectures are available, and most of them use convolutional neural networks (CNN, LeCun et al., 2015).

One of the successful object detection architectures is *You Only Look Once* (YOLO, Redmon et al., 2016), which is based on fully convolutional neural network architecture to predict bounding boxes and their class directly from whole input images. YOLO divides the image into a grid in which it indicates the boundary rectangles of objects and the probability of their classification. The main advantage of using YOLO is that we do not need any complex chaining of different processes for detection. After training the model on the input training data set, the algorithm can detect objects in real-time. Another advantage is that this architecture is invariant with respect to the size of the input image. Because YOLO can predict classes and bounding boxes in a single run of the algorithm, this method becomes incomparably faster than other algorithms designed to solve similar problems. The different applications prove that lower versions of YOLO also achieve excellent results in the detection of objects on spectrograms (Fanioudakis & Potamitis, 2017; Konan et al., 2020; O'Shea et al., 2017). Currently, the most recent version is YOLO version 5 (Jocher et al., 2020), which we decided to use for the processing of our data set. The referenced model we used for training and testing our lightning detection approach is YOLOv5s, which consists of 7.3 million parameters and 224 layers.

The YOLO network consists of three main parts: Backbone, Neck, and Head. The Backbone consists of a convolutional neural network that extracts features from images. YOLOv5 implements Cross Stage Partial Networks (CSP, Wang et al., 2020) for the CNN Backbone. CSP architectures are built on DenseNet (Huang et al., 2017), which suppresses the vanishing gradient problem (Hochreiter, 1998) and reduces the number of training parameters. Another subpart of the Backbone is also Spatial Pyramid Pooling (SPP, He et al., 2015), which eliminates the fixed-size constraints of the network. YOLOv5 implements the Path Aggregation Network (PANet, S. Liu et al., 2018) as the Neck for the aggregation of extracted features. PANet consists of a series of layers used to enhance the process of instance segmentation by preserving spatial information. The final output part is the Head which focuses primarily on detection—pre-

Table 1

For the Machine Learning Process, We Split Spectrogram Images From the Input Data Into Several Subsets

Set	Images	Events	Sferics	Tweeks
Training	1,566	11,244	9,691	1,553
Validation	396	2,767	2,355	412
Testing	219	2,135	1,890	245

Note. The displayed table provides the number of images and events (including the subtype numbers) selected in the training, validation, and testing subsets.

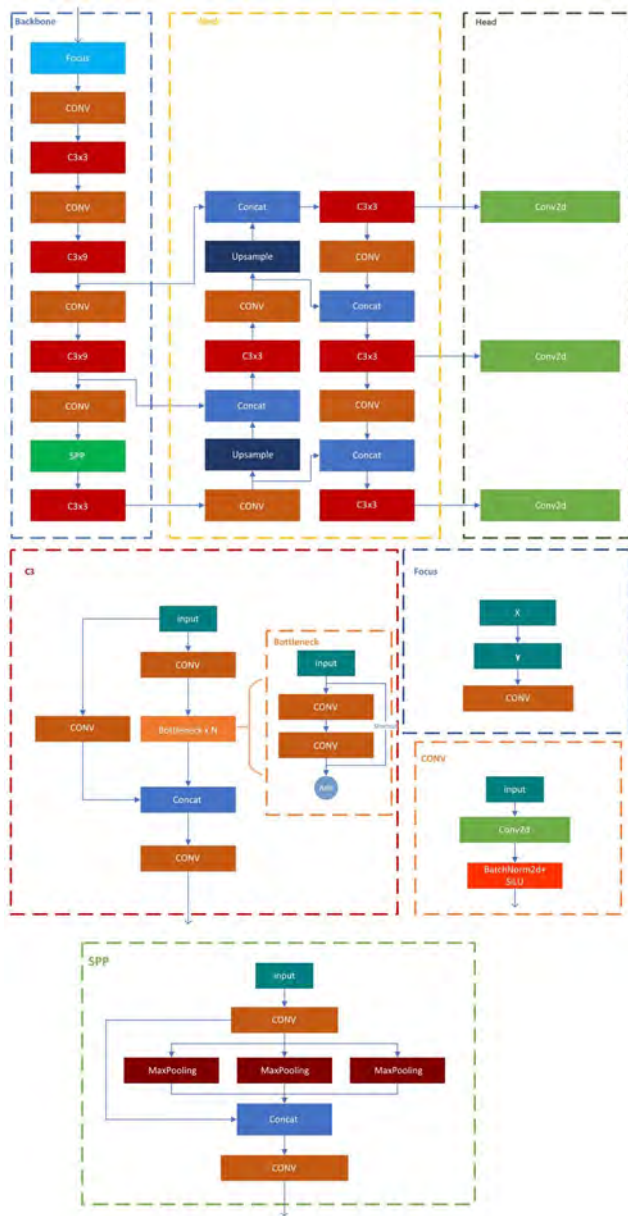


Figure 5. YOLOv5 architecture. The YOLO network consists of three main parts: Backbone, Neck, and Head displayed at the top part of the figure. Part Backbone and Neck use blocks displayed at the bottom part of the figure, where is described the architecture of blocks: Focus, CONV, C3 and SPP.

Table 2
Settings of the YOLOv5s Parameters for Training the Model

Hyper-parameters	Values
Image size	1184 × 160
Rectangular	true
Epochs	400
Batch size	4

dicting the bounding box and class. Figure 5 depicts the whole YOLOv5 architecture.

For the adaption of the YOLOv5s model for our task, we performed experiments with different setups and changes to the default model. Table 2 provides hyper-parameter settings used for the training process, and Figure 6 displays the learning process parameters and their changes during the training epochs. We reduced the size of the input image to 1184×160 pixels (input size must be a multiple of 32), and we set a rectangular inference. This value was optimal in the ratio of performance and hardware load. We used GPU Quadro RTX 4000 for the training of models. For traditional tasks, YOLO also offers the opportunity to use pre-trained models for classifying images with objects from the ordinary world. In our case, we did not use this option because we have specific inputs—spectrograms.

The YOLO algorithm strategy for detecting of bounding boxes is to divide each of the images into a grid, usually 19×19 . Based on this grid, each of the cells can predict a certain number of boxes. In order to detect bounding boxes, the YOLO network predicts them as deviations from a list of anchor box dimensions with the help of K-means and genetic algorithms introduced by Redmon and Farhadi (2018) in YOLO version 3. This part is crucial for solving tasks on custom datasets because the size distributions of our bounding boxes (narrow and long) are different from the box anchors in the original object detection data set. YOLO automatically re-learns anchor boxes from the new data set whenever we apply it on custom datasets. Subsequently, after the initial division of classes, it is necessary to perform another step known as Non-Maximum Suppression (NMS, Neubeck & Van Gool, 2006), which reduces boxes that are too close to each other based on the Intersection over Union—IoU metric (see Figure 7). This step removes boxes whose IoU is higher than the set IoU threshold. After the whole detection process, only the most accurate box for each object remains as output (Redmon et al., 2016).

There are three main settings for the detection phase in YOLO, one is the image size, and the other two are thresholds for the selection of bounding boxes. Table 3 displays our final setup of these parameters. For non-maximum suppression, we used the default IoU threshold value of 0.45 to reduce overlapping bounding boxes. However, we examined many experiments where we monitored the results based on the object confidence threshold. This value helped us to regulate the sensitivity of the detector, and it is a crucial setting. When the object confidence threshold value is small, the model detects more events. Figure 8 depicts such a case, where the object confidence threshold value was 0.1. On the other hand, the model could not detect positive events at a very high value. An example of setting the threshold to 0.45 is displayed in Figure 9. Based on several tests, we selected 0.3 as the optimal value for the object confidence threshold. Figure 10 provides an example of the prediction using this value of the threshold.

The setup of a threshold value for the detector's sensitivity can be seen as a user variable and is based on the expectations of a particular application. In our case, the threshold was selected towards the need to detect stronger signals in spectrograms. In other applications, the same model can detect weaker signals as detected events, using a lower threshold. Of

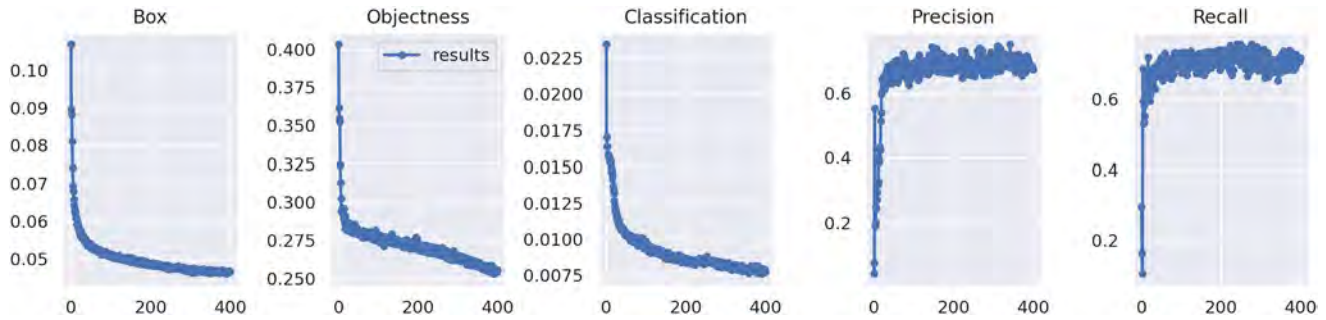


Figure 6. The plots display information on the learning process of our YOLOv5s model. The first three of them depict box, object, and class loss in individual epochs during training. The last two plots provide the evolution of precision and recall metrics. In all cases, the x-axis represents the number of epochs.

course, when very low thresholds are used, the problem with the false detections can arise more often due to the mixup with the background signal (low ratio between weak event and background signal).

The output of each detection of the YOLO algorithm consists of four values: the center of the box, the width and height of the box to delimit the detected object, the class of the predicted object, and the probability of prediction. In our case, every bounding box is a rectangle in spectrogram connected to a lightning event, which is a sferic or tweek.

3.2. Post-Processing

We used predictions from the YOLO network for two purposes—to provide basic information in the output table with detected events and as input for tasks to extract additional details of events. The output data table with detected events contains columns date and second extracted from the part of the jpg file name. We created column tweek based on the output of the neural network, which represents a binary value for whether the event is tweek (1) or sferic (0). For the detection of the precise time of event and occurrence of their emission under 2 kHz, it was necessary to use post-processing to obtain columns millisecond and $f_{\min} < 2\text{kHz}$. In both cases, we worked only with the red channel of the image for the decision.

First, the millisecond column contains the exact millisecond at which the lightning event occurred, represented by the imaginary center of the event. Since the most intensive part of the given event is not always in the middle of the prediction box (the highest intensity of tweeks is usually on the left from the middle of the box), it was necessary to work with the pixels and compare their intensity values within the prediction box. As mentioned above, we worked with the red channel of the separated

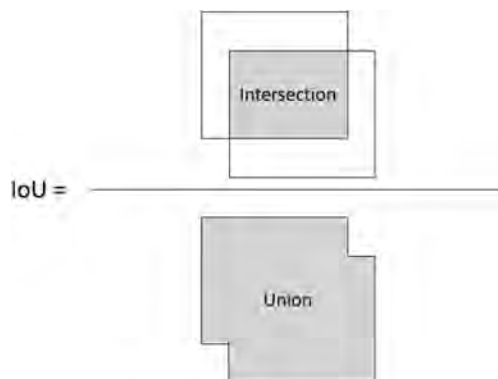


Figure 7. Intersection over Union.

bounding boxes of the events. We created a script that counts the sum of red pixels vertically over the entire height of the image (column of pixels, see Figure 11 for a better idea). The post-processing procedure compared the individual columns with each other, and the column with the highest value for the red channel identified the center of the event. After this step, we got the event position in pixel space. However, since the width of the whole spectrogram was 1421 pixels, to calculate its value in milliseconds, we still needed to divide the pixel value. Equation 1 shows the way to compute the exact value in milliseconds, where the value pixel of mid is the center position in pixels, 1,000 represents the number of milliseconds in one second, and width of spectrogram is the total width of the spectrogram, in our case 1421 pixels.

$$\text{millisecond} = \text{pixel of mid} * \frac{1000}{\text{width of spectrogram}} \quad (1)$$

Table 3
Setup of Input Parameters for the Detection Process

Parameter	Value
Image size	1184×160
NMS IoU threshold	0.45
Confidence threshold	0.30

We also needed to use a similar approach to provide information (for column $f_{\min} < 2\text{kHz}$) on whether the event has or does not have emission under 2 kHz. In this case, we did not work with the columns but with parts of the bounding box. The main idea is to compare emissions below 2 kHz with the emission of the entire image. In terms of dimensions, 2 kHz represents 20 bottom pixels of the bounding box. For each localized event, we summed the emission in the red channel on the bottom 20 pixels and compared it with the sum of all pixels of the bounding box.

We have compared these two values. We empirically found the optimal threshold for the comparison of emission of the subpart and the whole image. If the red channel emissions of the lower 20 pixels accounted for less than 9.5% of the red channels in the entire image, it was clear in most cases that this was not an event that reached below 2 kHz and the output value of $f_{\min} < 2\text{kHz}$ for this event in output table was 0. Otherwise, the output value was 1 representing that event has emission under 2 kHz. Figure 12 depicts a boundary of 2 kHz (as a black horizontal line) in one event example of the spectrogram.

The main output of the whole automatic detection process is in the form of a data table of detected events, which consists of the following columns extracted from predictions and post-processing steps:

1. image—represents the id number of the annotated spectrogram.
2. event—provides the sequence number of the localized event on the spectrogram.
3. date—is the extracted date and time of the measurement in the format YYYYMMDD_HHMMSS.
4. second—is the second of the 144-s measurement representing the interval of the spectrogram.
5. milisecond—represents the millisecond in which the event occurred, extracted by the method of automatic post-processing computation of the center of the event.
6. tweek—binary value, which represents whether the event is tweek (contains modes, value 1) or sferic (do not contain modes, value 0).
7. $f_{\min} < 2\text{kHz}$ —binary value, which provides information whether the event has emissions under 2 kHz in the spectrogram (based on the application of the post-processing method). If a given event reaches an area below 2 kHz, the value is 1; otherwise, the value is 0.

Figure 13 displays an example of an output table, which contains automatically detected events and is ready for subsequent analysis according to the needs of scientists in the area of atmospheric physics. The presented methods and the whole process were developed in the Python programming language (Van Rossum & Drake, 1995) using the Tensorflow (Abadi et al., 2016), and Keras (Chollet et al., 2015; Gulli & Pal, 2017) libraries. The source code, together with a detailed description, is available in the Jupyter notebook that forms the online material (Maslej-Krešňáková et al., 2021) for this article.

4. Method Evaluation and Discussion

In this section, we provide the evaluation of the detection process and its steps. It is important to note that there is always some annotation bias, that is, the annotators did not plot all bounding boxes perfectly. One of the advantages of neural network detection models is their ability to learn correct prediction functions from the annotation sets if most annotations are accurate. Therefore, the final model was able to suppress annotation errors, and the prediction boxes perfectly bounded the events. Hence, it is also important to perform a visual control of the boxes. The use of only quantitative comparison of predicted boxes and annotations might be insufficient.

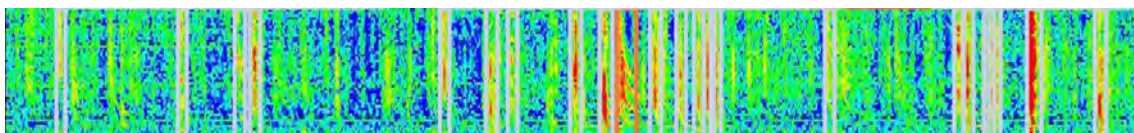


Figure 8. Example of detected bounding boxes with object confidence threshold 0.1. The model with this setting captures events with high sensitivity.

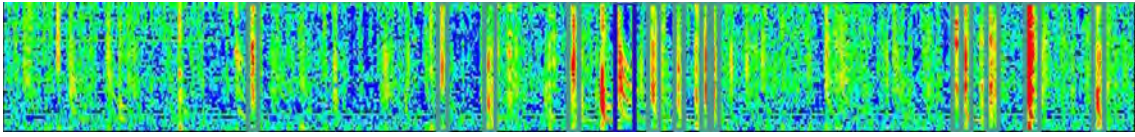


Figure 9. Example of detected events with object confidence threshold setting 0.45. In this case, the model did not capture multiple tweeks, so we did not test the higher threshold and looked for an optimal value between 0.25 and 0.45.

Since we chose the optimal value of the object confidence threshold as less than 0.50, there is a possibility that the model labels a given event as a sferic and a tweek simultaneously. Based on a visual inspection, we decided to create a script to evaluate the data to mark all duplicate predictions as a tweek. In most testing cases (113 out of 136), these events also had a higher value of the probability of predicting a tweek class than a sferic.

For the clarity of the analysis, the quantitative evaluation below does not include duplicate labels. We used a confusion matrix (Table 4) and metrics like precision, recall, and F1 score to evaluate the model using the test set. The main idea for comparison is to compare the same events from the annotation and detection output table. For this purpose, we obtained the identical event records by comparing the values of date and millisecond attributes from the output table and the annotation table.

The obtained confusion matrix is used to evaluate our detection model. Every element counts the number of cases that occurred. In general, we can say that each row of the confusion matrix represents the instances in a predicted class while each column represents the instances in an actual class. For example, as we can see in Table 4, the element TP represents all cases where our algorithm predicted the event as sferic (for Predicted = sferic in a row) and annotators labeled it also as sferic (for Actual = sferic in a column). The terms positive and negative in our case are used due to the standard terminology of the confusion matrix; let say positive detection is a prediction of sferic and negative detection is a prediction of tweek (i.e., not sferic). There are only two rows and two columns in a simple classification task (submatrix 2x2), but we need extra input on both axes for background values for the detection task. Therefore, we have Background False Negative values representing annotated events not found by the model (not matched to any event in the output table). Then we also have Background False Positive values representing the set of events found by the model but missing from the model annotations. So, the final confusion matrix contains the following elements:

- True Positive (TP)—the number of correctly detected sferics (all events both annotated and detected as sferic),
- True Negative (TN)—the number of correctly detected tweeks (all events both annotated and detected as tweek),
- False Positive (FP)—the number of events annotated as tweek but detected by the model as sferic,
- False Negative (FN)—the number of events annotated sferic but detected by the model as tweek,
- Background False Positive (BFP1, BFP2)—the number of events for the given class (1—sferic, 2—tweek) detected by the model, but not matched to any annotation,
- Background False Negative (BFN1, BFN2)—the number of events for the given class (1—sferic, 2—tweek) available in annotation table, but not detected by the model.

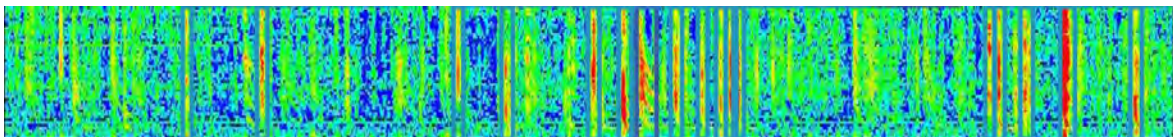


Figure 10. Example of detected bounding boxes with object confidence threshold setting 0.3. After several experiments, their quantitative evaluation, and visual inspection, we determined this value as optimal for the final setup of the YOLOv5 model applied to our data set.



Figure 11. An example of the event with the column of pixels representing the highest red value (black line)—extracted time of the event.

Table 5 provides precise numbers from the evaluation on testing set of 219 images. We used confusion matrix results to evaluate detection using metrics called precision, recall, and F1 score, for every class.

For the sferic class, precision and recall is calculated (based on the notation from Table 4) as follows:

$$\text{Precision(class = sferic)} = \frac{TP}{TP + FP + BFP1}, \quad (2)$$

$$\text{Recall(class = sferic)} = \frac{TP}{TP + FN + BFN1}. \quad (3)$$

For the tweek class, precision and recall are calculated as follows:

$$\text{Precision(class = tweek)} = \frac{TN}{TN + FN + BFP2}, \quad (4)$$

$$\text{Recall(class = tweek)} = \frac{TN}{TN + FP + BFN2}. \quad (5)$$

The F1 score is the harmonic mean of precision and recall. Therefore, we can compute it from the particular precision and recall values for every class using the formula:

$$\text{F1 score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (6)$$

Table 6 shows the evaluation of the model (with object confidence thresholds of 0.3) using precision, recall, and F1 score metrics. The column Images indicates the number of images on which we performed the prediction for both classes. The column Events shows the actual number of events from annotations. The Total row represents the summary for both classes—the number of all events and the weighted average of precision, recall, and F1 score from the values for particular classes, representing the overall prediction efficiency of our method compared to annotations.

The evaluation results show detection with an F1 score of ~70%, which is a good result for detection task based on manual annotations. The lower score is attributable to two circumstances. First, we had manual annotations, which always suffer from some annotation bias. As we wrote before, if most annotations are relatively precise, the model learns predictions better and suppresses the errors in annotation sets in detection. Of course, if we then compare original annotations with detections, the bias results in a lower evaluation score. Therefore, it is also essential to manually check some of the detections to see whether the predictions fulfill the expected needs. In particular, it was checked which events are detected in the spectrogram (according to predefined expectations), also if events have correct types (sferic or tweek), and whether extraction of additional features about events is correct (exact time of event, emissions under 2 kHz). The detailed manual check of more than 50 spectrograms proved that the current model provides output data in expected quality for further analysis and can be applied automatically on a larger data set of spectrograms.

During the evaluation, we also found another effect responsible for the lower metric values. Since the matching of bounding boxes from annotations and detections is based on the center of boxes (millisecond) obtained by the pixel values comparison described above, some centers may have been displaced by one pixel. Therefore, no match was found. In this case, the Background False Negative and Background False Positive values would increase directly. There is a possibility that different normalizations of the spectrogram can reduce



Figure 12. Example of an event that does not reach below 2 kHz. The black horizontal line represents the border for the algorithm to evaluate if events have an emission under 2 kHz.

image	event	date	second	millisecond	tweek	f_min<2kHz
8	1	20150902_231933	102	100	1	1
8	2	20150902_231933	102	129	1	1
8	3	20150902_231933	102	153	0	1
8	4	20150902_231933	102	190	1	1
8	5	20150902_231933	102	246	1	1
8	6	20150902_231933	102	417	0	0
8	7	20150902_231933	102	786	0	1
8	8	20150902_231933	102	813	0	1
8	9	20150902_231933	102	864	0	1
8	10	20150902_231933	102	913	0	0
8	11	20150902_231933	102	966	0	1

Figure 13. The figure provides an example of the output table, which results from applying our automatic detection method to spectrograms for the lightning events detection. The data prepared in this way can now be used for further analysis and statistical processing.

noise and provide a more precise computation of the time of the event. Also, we can use more relaxed matching for events between annotation and detections (1–2 pixels). However, the results of the YOLO model will not change, and from the current evaluation (both quantitative and qualitative—manual visual control), we assume that predicted bounding boxes are of high quality. Therefore, our approach provides a valuable tool for automated detection of lightning events in spectrograms from the ELMA-VAN-G analyzer.

We also evaluated the correctness of the automatic method for the identification of events with emissions below 2 kHz (column f_min<2kHz in the output table). For the evaluation, we manually annotated 1,967 events (bounding boxes) for the testing set. In this case, we have a simple binary classification task, with two classes—over 2 kHz and below 2 kHz. Table 7 presents the confusion matrix, with the values of TP, TN, FP, FN. The meaning of the values is similar to the previous case, just simplified for binary classification. Therefore, the calculations of precision and recall for particular classes are as follows (based on the notation available in 7):

$$\text{Precision(class = over 2 kHz)} = \frac{TP}{TP + FP}, \quad (7)$$

$$\text{Recall(class = over 2 kHz)} = \frac{TP}{TP + FN}, \quad (8)$$

$$\text{Precision(class = below 2 kHz)} = \frac{TN}{TN + FN}, \quad (9)$$

$$\text{Recall(class = below 2 kHz)} = \frac{TN}{TN + FP}, \quad (10)$$

Table 8 displays the results of this evaluation. Similar to the previous evaluation, the F1 score is computed as the harmonic mean of particular precision and recall for every class. Also, the total value is a summary for both classes computed as a weighted average, and the Events column contains the number of events in a class (or total). We can see that 79 events were misclassified as events that didn't reach below 2 kHz, that is, false positive and 42 false negative events. According to the results, this postprocessing step showed very good performance with an overall 94% F1 score for the prediction.

5. Conclusions

We have developed a reliable automatic method for the extraction of the required details from frequency-time spectrograms. The presented method is based on a deep learning and deterministic approach. Thanks to the YOLOv5 algorithm, we can detect sferics and tweeks from spectrograms and, in post-processing,

Table 4
Confusion Matrix—Selected Notation of Elements Representing Different Cases for the Detection Task

Actual				
Predicted		Sferic	Tweek	Background
	Sferic	TP	FP	BFP1
	Tweek	FN	TN	BFP2
	Background	BFN1	BFN2	

Table 5
The Table Presents a Confusion Matrix Used to Evaluate Our Event Detection Method on the Annotated Test Set

Actual				
Predicted		Sferic	Tweek	Background
	Sferic	1,278	71	379
	Tweek	62	138	39
	Background	550	36	

Table 6
The Table Displays the Evaluation of the Detection Model (With the Object Confidence Threshold of 0.3) Using Metrics Like Precision, Recall, and F1 Score

Class	Images	Events	Precision	Recall	F1 score
Sferic	219	1,890	0.74	0.68	0.71
Tweek	219	245	0.58	0.56	0.57
Total	219	2,135	0.72	0.66	0.69

Note. The values for Total represent the weighted average of the metrics.

Table 7
The Table Presents a Confusion Matrix Representing the Evaluation of the Automatic Method for Identifying Events With an Emission Below or Over 2 kHz

Actual				
Predicted		Over 2 kHz	Below 2 kHz	
	Over 2 kHz	TP = 1,011	FP = 79	
	Below 2 kHz	FN = 42	TN = 835	

Table 8
The Table Provides Evaluation Metrics for the Automatic Method for the Identification of Emission Below or Over 2 kHz

Class	Events	Precision	Recall	F1 score
Over 2 kHz	1,053	0.93	0.96	0.94
Below 2 kHz	914	0.95	0.91	0.93
Total	1,967	0.94	0.94	0.94

Note. The values for the class Total represent the weighted average of the precision, recall, and F1 score per class. The Events column contains the number of events per class or total.

determine the millisecond in which the event occurred and provides information on whether the event has emissions under 2 kHz. For the training and testing process, we annotated more than 2,000 spectrograms with the help of students using an online annotation platform. The annotation set finally provided 3,200 tweeks and 19,000 sferics, which helped train the YOLO model. The presented method can now extract details about detected lightning events and store them in the form of a table for further statistical scientific analysis. The proposed method is very fast and also suitable for real-time deployment. The current model was trained to provide results for data recorded by the ELMAVAN-G instrument, but thanks to the transfer learning ability of deep learning approach, the same architecture can be re-trained for the application to a new data set. Therefore, the whole process of spectrogram data processing, from identification of sources, annotations, the learning phase, and the use of the trained model, can also be seen as a methodology or a use-case scenario for similar applications in the future.

After normalizing the data, we expect that the noise will be reduced, and the method will be able to extract details from spectrograms with even higher accuracy. The trained model can also be used to detect low or, on the contrary, only extreme events. The sensitivity of the detector can be regulated based on the setting of the detection threshold. It is also possible to detect only one selected class. The time of occurrence and the frequency extent of individual sferics, a presence/absence of the signs of dispersion, as well as the presence of higher tweek harmonics represent valuable information which can be used for routine characterization of the ionospheric D-layer. The characteristics of different waveguide modes, as well as the influence of solar events on the occurrence of sferics and their propagation in the Earth-ionosphere waveguide can be derived from the information obtained using the presented machine learning technique. This will also lead to a better understanding of the effects of the terrestrial magnetic field, and meteorological or seismogenic events on the propagation of sferics and on the properties of the D-layer. Therefore it might significantly contribute also to the study of thermosphere-ionosphere variations presented by Mackovjak et al. (2021). Even more, such extensive analysis of several years of VLF data has not been performed yet, and could bring interesting information about diurnal and seasonal variability of signal propagation effects, as well as their dependence on different phases of the solar cycle. Results can also serve as an input for the models of the density profile of the ionospheric D-layer or for VLF propagation codes.

Data Availability Statement

The data were provided by I. Kolmašová and O. Santolík and are publicly available through the database of the Institute of Atmospheric Physics of the Czech Academy of Sciences (<http://bleska.ufa.cas.cz/labb/storage/elm/>). The presented results can be reproduced by the Jupyter notebook publicly available at <https://doi.org/10.5281/zenodo.5494705>.

Acknowledgments

The studies of the deep learning approach presented in this work are supported by Slovak APVV research grant under contract No. APVV-16-0213 and Slovak VEGA research grant No. 1/0685/21. I. Kolmašová and O. Santolík acknowledge support from the Czech Grant Agency project no. 20-09671S and by the European Regional Development Fund-Project CREAT (CZ.02.1.01/0.0/0.0/15_003/0000481). Š. Mackovjak acknowledges support by the Slovak VEGA grant 2/0155/18 and by the government of Slovakia through ESA contracts No. 4000125330/18/NL/SC and No. 4000125987/18/NL/SC under the PECS. ESA Disclaimer: The view expressed herein can in no way be taken to reflect the official opinion of the European Space Agency. We are thankful to students of the Technical University of Košice who helped us with the preparation of Custom annotations using the Zooniverse.org platform. This publication uses data generated via the Zooniverse.org platform, the development of which is funded by generous support, including a Global Impact Award from Google, and by a grant from the Alfred P. Sloan Foundation.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th usenix symposium on operating systems design and implementation (osdi 16)* (pp. 265–283).
- Bernard, L. (1973). A new nose extension method for whistlers. *Journal of Atmospheric and Terrestrial Physics*, 35(5), 871–880. [https://doi.org/10.1016/0021-9169\(73\)90069-x](https://doi.org/10.1016/0021-9169(73)90069-x)
- Brabham, D. C. (2008). Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1), 75–90. <https://doi.org/10.1177/1354856507084420>
- Budden, K. G. (1961). *The wave-guide mode theory of wave propagation*. Logos Press.
- Chang, J.-W., Kang, K.-W., & Kang, S.-J. (2019). Sdnn: An efficient sparse deconvolutional neural network accelerator on fpga. In *2019 design, automation & test in europe conference & exhibition (date)* (pp. 968–971). <https://doi.org/10.23919/date.2019.8715055>
- Chollet, F., et al. (2015). Keras. *GitHub*. Retrieved from https://keras.io/getting_started/faq/#how-should-i-cite-keras
- Christian, H. J., Blakeslee, R. J., Boccippio, D. J., Boeck, W. L., Buechler, D. E., Driscoll, K. T., & Stewart, M. F. (2003). Global frequency and distribution of lightning as observed from space by the optical transient detector. *Journal of Geophysical Research*, 108(D1). ACL 4-1-ACL 4-15. <https://doi.org/10.1029/2002JD002347>
- Chronis, T., Carey, L. D., Schultz, C. J., Schultz, E. V., Calhoun, K. M., & Goodman, S. J. (2015). Exploring lightning jump characteristics. *Weather and Forecasting*, 30(1), 23–37. <https://doi.org/10.1175/waf-d-14-00064.1>
- Cummer, S. A., Inan, U. S., & Bell, T. F. (1998). Ionospheric d region remote sensing using vlf radio atmospherics. *Radio Science*, 33(6), 1781–1792. <https://doi.org/10.1029/98RS02381>
- Deierling, W., & Petersen, W. A. (2008). Total lightning activity as an indicator of updraft characteristics. *Journal of Geophysical Research*, 113(D16), D16210. <https://doi.org/10.1029/2007JD009598>
- Fanioudakis, L., & Potamitis, I. (2017). *Deep networks tag the location of bird vocalisations on audio spectrograms*. arXiv preprint arXiv:1711.04347.
- Grubor, D. P., Šulić, D. M., & Žigman, V. (2008). Classification of x-ray solar flares regarding their effects on the lower ionosphere electron density profile. *Annales Geophysicae*, 26(7), 1731–1740. <https://doi.org/10.5194/angeo-26-1731-2008>
- Gulli, A., & Pal, S. (2017). *Deep learning with keras*. Packt Publishing Ltd.
- Haldoupis, C., Cohen, M., Arnone, E., Cotts, B., & Dietrich, S. (2013). The vlf fingerprint of elves: Step-like and long-recovery early vlf perturbations caused by powerful cg lightning em pulses. *Journal of Geophysical Research: Space Physics*, 118(8), 5392–5402. <https://doi.org/10.1002/jgra.50489>
- Han, F., & Cummer, S. A. (2010). Midlatitude nighttime d region ionosphere variability on hourly to monthly time scales. *Journal of Geophysical Research*, 115(A9), A09323. <https://doi.org/10.1029/2010JA015437>
- Harid, V., Liu, C., Pang, Y., Alvina, A. J., Golkowski, M., Hosseini, P., & Cohen, M. (2021). Automated large-scale extraction of whistlers using mask-scoring regional convolutional neural network. *Geophysical Research Letters*, 48(15), e2021GL093819. <https://doi.org/10.1029/2021GL093819>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904–1916. <https://doi.org/10.1109/tpami.2015.2389824>
- Helliwell, R. (1965). *Whistlers and related ionospheric phenomena*. Stanford University Press.
- Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(2), 107–116. <https://doi.org/10.1142/s0218488598000094>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708). <https://doi.org/10.1109/cvpr.2017.243>
- Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2018). *Densely connected convolutional networks*.
- Huffines, G. R., & Orville, R. E. (1999). Lightning ground flash density and thunderstorm duration in the continental United States: 1989–96. *Journal of Applied Meteorology*, 38(7), 1013–1019. [https://doi.org/10.1175/1520-0450\(1999\)038<1013:lgfdat>2.0.co;2](https://doi.org/10.1175/1520-0450(1999)038<1013:lgfdat>2.0.co;2)
- Inan, U. S., Cummer, S. A., & Marshall, R. A. (2010). A survey of elf and vlf research on lightning-ionosphere interactions and causative discharges. *Journal of Geophysical Research*, 115(A6), A00E36. <https://doi.org/10.1029/2009JA014775>
- Jacobson, A. R., Holzworth, R. H., & Brundell, J. B. (2021). Using the world wide lightning location network (wwlln) to study very low frequency transmission in the earth-ionosphere waveguide: 1. comparison with a full-wave model. *Radio Science*, 56(7), e2021RS007293. <https://doi.org/10.1029/2021RS007293>
- Jocher, G., Stoken, A., Borovec, J., NanoCode012., Christopher, S., Changyu, L., & Rai, P. (2020). *ultralytics/yolov5: v3.1 - Bug fixes and performance improvements*. Zenodo. <https://doi.org/10.5281/zenodo.4154370>
- Kolmašová, I., Santolík, O., Kašpar, P., Popek, M., Pizzuti, A., Spurný, P., et al. (2021). First observations of elves and their causative very strong lightning discharges in an unusual small-scale continental spring-time thunderstorm. *Journal of Geophysical Research: Atmospheres*, 126(1), e2020JD032825. <https://doi.org/10.1029/2020JD032825>
- Konan, O. J., Mishra, A. K., & Lotz, S. (2020). *Machine learning techniques to detect and characterise whistler radio waves*. arXiv preprint arXiv:2002.01244.
- Kumar, S., Kishore, A., & Ramachandran, V. (2008). Higher harmonic tweek sferics observed at low latitude: Estimation of vlf reflection heights and tweek propagation distance. *Annales Geophysicae*, 26(6), 1451–1459. <https://doi.org/10.5194/angeo-26-1451-2008>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lichtenberger, J., Ferencz, C., Bodnár, L., Hamar, D., & Steinbach, P. (2008). Automatic whistler detector and analyzer system: Automatic whistler detector. *Journal of Geophysical Research*, 113(A12), A12201. <https://doi.org/10.1029/2008ja013467>
- Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128(2), 261–318. <https://doi.org/10.1007/s11263-019-01247-4>
- Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8759–8768). <https://doi.org/10.1109/cvpr.2018.00913>
- Mackovjak, S., Varga, M., Hrivnak, S., Palkoci, O., & Didebulidze, G. G. (2021). Data-driven modeling of atomic oxygen airglow over a period of three solar cycles. *Journal of Geophysical Research: Space Physics*, 126(3), e2020JA028991. <https://doi.org/10.1029/2020JA028991>
- Maslej-Krešňáková, V., Kunderát, A., Mackovjak, S., Butka, P., Jaščur, S., Kolmašová, I., & Santolík, O. (2021). *space-lab-sk/tweaks_detection: First release*. Zenodo. <https://doi.org/10.5281/zenodo.5494705>
- Maurya, A. K., Singh, R., Veenadhari, B., Kumar, S., Cohen, M. B., Selvakumaran, R., et al. (2012). Morphological features of tweeks and nighttime d region ionosphere at tweek reflection height from the observations in the low-latitude indian sector. *Journal of Geophysical Research*, 117(A5), A05301. <https://doi.org/10.1029/2011JA016976>

- Mezentsev, A., Lehtinen, N., Østgaard, N., Pérez-Invernón, F. J., & Cummer, S. A. (2018). Spectral characteristics of vlf sferics associated with rhesi tgfs. *Journal of Geophysical Research: Atmospheres*, 123(1), 139–159. <https://doi.org/10.1002/2017JD027624>
- Mogilevsky, M., Zelenyi, L., Demekhov, A., Petrukovich, A., & Shklyar, D. (2012). Resonance project for studies of wave-particle interactions in the inner magnetosphere. *American Geophysical Union*, 199, 117–126.
- Neubeck, A., & Van Gool, L. (2006). Efficient non-maximum suppression. In *18th international conference on pattern recognition (icpr'06)* (Vol. 3, pp. 850–855). <https://doi.org/10.1109/icpr.2006.479>
- Ohya, H., Nishino, M., Murayama, Y., & Igarashi, K. (2003). Equivalent electron density at reflection heights of tweek atmospherics in the low middle latitude d-region, ionosphere. *Earth Planets and Space*, 55, 627–635. <https://doi.org/10.1002/2014JA020375>
- Ohya, H., Shiokawa, K., & Miyoshi, Y. (2015). Daytime tweek atmospherics. *Journal of Geophysical Research: Space Physics*, 120(1), 654–665. <https://doi.org/10.1002/2014JA020375>
- O'Shea, T., Roy, T., & Clancy, T. C. (2017). Learning robust general radio signal detection using computer vision methods. In *2017 51st asilomar conference on signals, systems, and computers* (pp. 829–832).
- Piša, D., Němec, F., Santolik, O., Parrot, M., & Rycroft, M. (2013). Additional attenuation of natural vlf electromagnetic waves observed by the demeter spacecraft resulting from preseismic activity. *Journal of Geophysical Research: Space Physics*, 118(8), 5286–5295. <https://doi.org/10.1002/jgra.50469>
- Price, C., Asfur, M., & Yair, Y. (2009). Maximum hurricane intensity preceded by increase in lightning frequency. *Nature Geoscience*, 2(5), 329–332. <https://doi.org/10.1038/ngeo477>
- Pulinets, S., & Boyarchuk, K. (2005). *Ionospheric precursors of earthquakes*. Springer.
- Rapoport, Y., Grimalsky, V., Fedun, V., Agapitov, O., Bonnell, J., Grytsai, A., et al. (2020). Model of the propagation of very low-frequency beams in the earth–ionosphere waveguide: Principles of the tensor impedance method in multi-layered gyrotropic waveguides. *Annales Geophysicae*, 38(1), 207–230. <https://doi.org/10.5194/angeo-38-207-2020>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 779–788). <https://doi.org/10.1109/cvpr.2016.91>
- Redmon, J., & Farhadi, A. (2018). *Yolov3: An incremental improvement*. arXiv preprint arXiv:1804.02767.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). https://doi.org/10.1007/978-3-319-24574-4_28
- Said, R. K., Inan, U. S., & Cummins, K. L. (2010). Long-range lightning geolocation using a vlf radio atmospheric waveform bank. *Journal of Geophysical Research*, 115(D23), D23108. <https://doi.org/10.1029/2010JD013863>
- Santolik, O., & Kolmašová, I. (2017). Unusual electromagnetic signatures of European North Atlantic winter thunderstorms. *Scientific Reports*, 7(1), 1–9. <https://doi.org/10.1038/s41598-017-13849-4>
- Schumann, U., & Huntrieser, H. (2009). The global lightning-induced nitrogen oxides source. *Atmospheric Chemistry and Physics*, 7(14), 3823–3907. <https://doi.org/10.5194/acp-7-3823-2007>
- Tan, L. M. (2016). Investigation of the morphology and wait's parameter variations of the low-latitude d region ionosphere using the multiple harmonics of tweeks. *Advances in Space Research*, 57(12), 2444–2451. <https://doi.org/10.1016/j.asr.2016.03.030>
- Thomson, N. R., & McRae, W. M. (2009). Nighttime ionospheric d region: Equatorial and nonequatorial. *Journal of Geophysical Research*, 114(A8), A08305. <https://doi.org/10.1029/2008JA014001>
- Van Rossum, G., & Drake, F. L., Jr (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam.
- Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops* (pp. 390–391). <https://doi.org/10.1109/cvprw50498.2020.00203>
- Williams, E., Rothkin, K., Stevenson, D., & Boccippio, D. (2000). Global lightning variations caused by changes in thunderstorm flash rate and by changes in the number of thunderstorms. *Journal of Applied Meteorology*, 39(12), 2223–2230. [https://doi.org/10.1175/1520-0450\(2001\)040<2223:glvcbc>2.0.co;2](https://doi.org/10.1175/1520-0450(2001)040<2223:glvcbc>2.0.co;2)
- zooniverse.org. (2021). Retrieved from <https://www.zooniverse.org>