Polyphonic Bird Sound Event Detection With Convolutional Recurrent Neural Networks

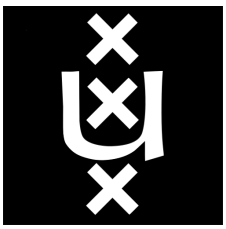
SUBMITTED IN PARTIAL FULLFILLMENT FOR THE DEGREE OF MASTER OF SCIENCE

Maximilian Crous
10771085

MASTER INFORMATION STUDIES
Information Systems

FACULTY OF SCIENCE
UNIVERSITY OF AMSTERDAM

18th of July



1st Examiner
dhr. dr. Arnoud Visser
Faculty of Science, Informatics Institute

2nd Examiner
dhr. dr. Ashley Burgoyne
Faculty of Humanities, Institute for Logic, Language and Computation

Polyphonic Bird Sound Event Detection with Convolutional Recurrent Neural Networks

Maximilian Crous
Faculty of Science

Faculty of Natural Sciences, Mathematics and Computer Science

University of Amsterdam
Science Park, Amsterdam, Netherlands
max.crous@student.uva.nl

***Abstract*—This thesis investigates the use of machine learning for predicting the onset, duration and species of overlapping bird sounds in field recordings. The resulting model - a modified version of SEDnet - achieves state-of-the-art performance with a frame-wise F-score of 0.94 and an error rate of 0.11 on recordings with up to three overlapping sound sources. It is further shown that the features learned by the model have the capacity to segment bird songs into song phrases. To train and test the model, a novel bird sound dataset was created with 500+ sound event annotations for each of 5 bird species. The model, code and dataset are publicly available[1].**

***Index Terms*—Sound Event Detection, Birds, Biomonitoring, Species Identification**

I. Introduction

Birds are an exceptional group of animals, in that they broadcast their presence with their songs and calls. This trait offers great potential for biomonitoring, as audio can be used to confirm bird presence at a range of distances whilst not requiring the animal to be visible. Despite this, bird population estimation is currently performed by manual visual surveying and quadrat sampling, often including volunteers to help address the challenges of scale [1] [2]. To put audio analysis forward as a viable alternative for bird population estimation requires advancements in audio analysis and annotated bird sound datasets. This work addresses both issues by proposing a polyphonic sound event detection (PSED) model and a time annotated bird sound dataset. The proposed model consists of a convolutional neural network (CNN) linked to a bidirectional recurrent layer, with each output neuron predicting the classes present in a single audio frame. The dataset contains over 3200 annotations, with over 500 annotations for blackbirds, chiffchaffs, great tits, warblers and wrens. Apart from the PSED model, this paper delves into the process of creating the dataset, segmenting bird song phrases by clustering CNN features, and possible future use cases for the model.

[1]https://github.com/maxcrous/bsed

II. Theoretical Background

A. Information Stored In Bird Songs

A bird vocalization can be either a song or a call. In general, songs tend to be long and complex and serve territorial and mating functions. Conversely, calls are simpler, occur less spontaneously and are used for contacting and warning conspecifics [3]. Studies have shown that songs can reveal a number of things about the birds singing them. Certain types of songs are perceived as territorial threats [4], with greater song complexity adding to the perceived threat [5]. For several species, the size of song repertoires correlates with fitness [6] and songs from adults can bear the signs of nutritional stress experienced during early development [7]. Even parasitic infections have been shown to decrease birds' song rate and weaken their frequency range [8] [9]. Lastly, the set of songs sung by a bird can be codetermined by its social environment, as both great tits and sparrows have shown repertoire matching between neighbouring conspecifics [10] [11]. Thus, the information stored in bird songs goes well beyond the singer's species or the number of birds present in a scene. These findings set the precedent for the kind of information we may wish to automatically extract with audio analysis and hints at why detecting birds in recordings could be valuable.

B. Bird Sound datasets

Supervised machine learning models require labelled data in order to distinguish between various categories and detect patterns. The design and format of a dataset depends on the nature of the supervised learning task.

1) Presence Prediction: For the task of bird presence prediction, several datasets exist. One that stands out with respect to scale is the conglomerate Warblr dataset [12]. It consists of 32000 audio files, each 10 seconds long and annotated with either a bird *present* or *absent* tag. The dataset combines the Warblr project's in-house data, the freefield dataset [13] and the BirdVox-DCASE-20k dataset [14].

2) Species Classification: The largest publicly available collection of bird sounds with species annotations is the crowd

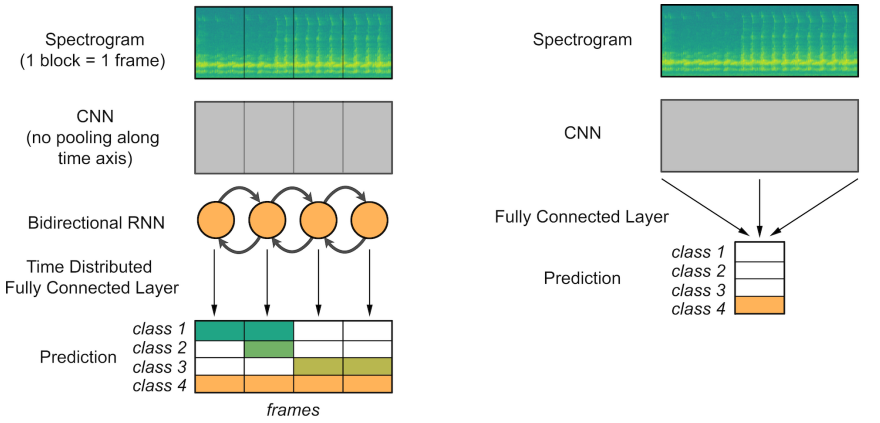


Fig. 1: Model architectures

sourced Xeno-canto repository [15]. It houses over 470.000 recordings covering over 10.000 species[2]. The lengths of the recordings range from a few seconds to over an hour. Each recording contains rich metadata, such as the recording location, time of day and a user description. An especially important metadata attribute for this work is the recording rating. Each recording on Xeno-canto has a crowd sourced rating that ranges from *A - Loud and Clear* to *E - Barely audible*[3]. While Xeno-canto is not a readily usable dataset in itself, the repository has served as a source of data for the BirdCLEF species classification dataset since 2014 [16]. The most recent 2019 BirdCLEF[4] dataset covers 659 common species from North and South America, with each species being featured in at least 15 and at most 100 audio files.

3) Bird Sound Event Detection: A supervised model trained to detect the onset and duration of sound events requires temporal annotations. To the best of our knowledge, there are only two substantial bird sound datasets with time annotations: the 2019 BirdCLEF soundscapes [16] and the BirdVox-full-night dataset [17]. Both have shortcomings that rendered them unusable for this work. The BirdVox dataset is species-agnostic and only contains the onset of each sound, but not the duration or offset. The BirdCLEF soundscapes did not contain specific onset and offset time annotations. Instead, the recordings were split into 5 second segments for which the occurring species were annotated. It is important to note that fixed length 5 second annotations are not specific enough to train the millisecond precise model presented in this work. While datasets with precise time annotations were not found for birds, they do exist for other classes. For instance, the DCASE Task 5 dataset [18] contains time annotated sound events for cooking, dishwashing, social activity and more. The lack of such a dataset for birds motivated the creation of the dataset used in this paper.

[2]https://www.Xeno-canto.org/collection/stats/graphs
[3]https://www.Xeno-canto.org/help/FAQrating
[4]https://www.imageclef.org/BirdCLEF2019

C. Bird Recognition with Convolutional Neural Networks

Attempts to predict bird presence and classify bird species within fixed length recordings have achieved considerable success, partly due to the yearly challenges hosted by BirdCLEF [16] and Warblr [12]. The top entries in these competitions transform audio data to frequency domain representations, interpret these representations as images (spectrograms), and apply CNN classification [19] (see Fig 1b). These networks are highly effective, achieving an AUC of up to 89% on bird presence prediction [20] and 0.83% MRR on species classification [21].

Despite the accuracy of these models, their utility for biomonitoring is limited due to several factors:

- The models predict a bird class for the whole recording. Information regarding the onset and offset of a bird sound can only be implicitly inferred and requires supplementary tools, such as Class Activation Maps [22] and Saliency Maps [23]. This means that a biological monitor will not be able diagnose which part of the recording gave rise to a certain classification, which is valuable information.
- There has been little research on the use of these models for multi-label classification, even though there are often multiple bird species present within a single field recording. To the best of our knowledge, there has only been a single published work on the performance of multi-label bird classification with CNNs, which was implicitly measured by mean reciprocal rank [21]. That is to say, the predictions were ranked in order of prediction confidence and the models were not trained to explicitly output multiple classes.

These issues are addressed by extending this type of classification model to a sound event detection model.

D. Models for Polyphonic Sound Event Detection

Sound event detection is the task of detecting the type, starting time, and ending time of sound events. In the polyphonic case, sound events can temporally overlap with each other

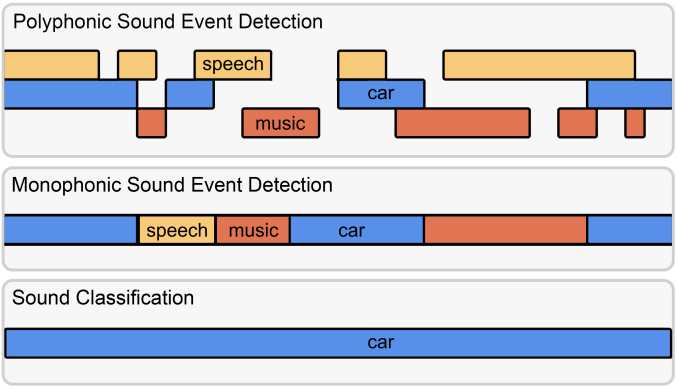


Fig. 2: Sound recognition tasks, edit of on image from [24]

(see Fig 2). The first published attempts at solving PSED employed chains of classical machine learning algorithms [25] [26] [27]. For instance, the authors of [25] used non-negative matrix factorisation to separate a spectrogram into its constituent sources. Thereafter, they used the viterbi algorithm to detect events and classes within those sources. This was made possible by interpreting the separated sound sources as hidden markov models, where the spectrograms were the visible variables and the sound events were the latent variables.

After the advent of deep learning [28], research into PSED moved away from classical machine learning algorithms and works that employed deep neural networks started to appear [29] [30] [31]. The neural network architecture that is used in this work was inspired by SEDnet [31]. SEDnet and other networks like it are called recurrent convolutional neural networks (CRNNs). The CRNN in this work uses a CNN to extract local features for each time step of a recording. Here, a time step is defined as a single unit along the time axis in the spectrogram of a recording (see top of Fig 1a). The model is able to maintain a feature for each time step by not pooling along the time axis. After the CNN module, a recurrent neural network [32] spreads the local features from a single time step to its temporal surroundings. This spreading of information with recurrent layers has multiple benefits:

- The network can make better informed predictions, as multiple similar sounds will reinforce a prediction. E.g. a feature that by itself equally supports a sparrow and a blackbird prediction, will skew towards predicting a blackbird if the preceding and following feature predict a blackbird.
- The network can be trained to include brief moments of silence in a prediction. For example, a *human speech* event contains many short silences in between words, which we may want to include in the prediction of a *human speech* event.

After the recurrent layers, the spread features are processed by a time distributed fully connected layer to make multi-label predictions for each time step. For a simplified visualization of the model, see Fig 1a.

III. Methods

A. Creating the Dataset

There are 2 popular approaches for annotating PSED datasets [24]. The first involves annotating audio that contains a mixture of sources. While mixed source audio is the most truthful representation of polyphonic scenes, annotating it is very time consuming and requires a highly perceptive annotator. In the case of birds sounds, it would imply that the annotator is able to recognise a large number of bird species and precisely determine song on- and offset in cluttered scenes.

The second method involves creating synthetic mixtures by combining the audio and annotations of isolated sound events. This allows the annotator to focus on annotating a single class for an extended period of time. The task thus becomes less mentally taxing as the class is more easily recognised and event on- and offsets are more easily determined. The dataset in this work is a collection of isolated event annotations.

1) Collecting Audio: All audio files that are included in the dataset in this work were downloaded in batches from Xeno-canto. First, a search query was issued on Xeno-canto for each species. The query flags filtered the results down to A rated recordings of bird songs. Any lower rated recordings or recordings of other vocalisations, such as fighting calls, were discarded. The species were chosen based on their prevalence in the Netherlands and number of Xeno-canto results. This was done in the hopes of using trained models on Dutch field recordings. The only species that does not appear in the 10 most common birds in the Netherlands [33] is the warbler. Querying and batch downloading was achieved by refactoring and running a Python scraping script named Birdbrain[5]. Once downloaded, each audio file was prepared for annotating by being split into sequences of 20 seconds with FFmpeg[6].

2) Annotations: Annotations were made within an interactive web application that allows users to brush select parts of a recording and assign classes (see Fig 6). The frontend was an unmodified copy of the CrowdCurio audio-annotator[7].

[5]https://github.com/davipatti/birdbrain
[6]https://ffmpeg.org
[7]https://github.com/CrowdCurio/audio-annotator

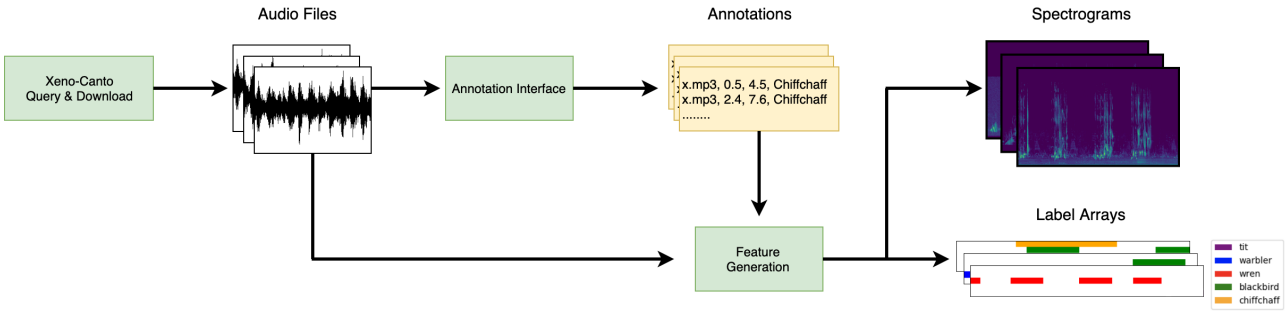


Fig. 3: Dataset creation pipeline

The backend was based on the Dynitag project[8]. Dynitag was originally programmed as a collaborative annotation tool that could be accessed through the internet. It was modified for this work to only allow access to a single local user. Instructions on how to set up this annotation environment can be found in this work's codebase.

Although all recordings were A rated, many contained faint traces of birds singing in the background. To reduce false negatives in the data, a special *not right* tag was used to disregard parts of the recordings (see Fig 7).

The design of the annotation storage format in this work was inspired by the format of the TUT Sound events 2017 development dataset [34]. In it, each sound event annotation is a line in a text file. Each line has the following format:

$$audio\_file\_path,\ event\_onset,\ event\_offset,\ class$$

An example entry would be:

$$BirdSounds1.mp3,\ 10.2,\ 14.9,\ Chiffchaff$$

This design was chosen due to its simplicity.

B. From Annotations to Features

1) Short-time Fourier transform: The PSED model in this work uses spectrograms to make predictions. A spectrogram is a time-frequency transform of a signal. There is no one-to-one mapping from audio files to spectrograms, as there are several different algorithms that can produce spectrograms, each of which has an array of hyperparameters. This work uses the short-time Fourier transform (STFT) to generate spectrograms, as it has been used in many other works on bird sound recognition [12] [16] [20]. One of the STFT's principal hyperparameters is its *window length*. This parameter determines how many audio wave sample points constitute a single spectrogram frame. It thus encodes a tradeoff between time resolution and frequency resolution. The best value for this parameter depends on the task at hand and would ideally be determined empirically by training a model with different values. To the best of the authors knowledge, there has been no such work that thoroughly reviews this parameter for bird songs, despite some works experimenting with up to three different values [35]. Due to the computational costs of training a PSED model, this work also does not provide such a review. Instead, we chose to manually evaluate 15 different window lengths, ranging from 16 frames to 2560 frames, for 100 different spectrograms (see Fig 8 for an excerpt of this comparison). The window length that on average revealed the structure of bird song most clearly was 512 frames. This window length was used throughout the work.

2) Sequence Length: Once a spectrogram has been created for every audio file in the database, a sequence length needs to be selected. The sequence length is the number of spectrogram frames that the PSED model will receive as input. In other words, the sequence length is proportional to the amount time or temporal context that the network can take into consideration when making predictions. Sequence length thus encodes a tradeoff between a model's memory efficiency and the amount of information contained in each sample.

In the Warblr bird presence prediction challenge, contestants mostly used 10 second sequence lengths [12] [20], as that matched the 10 second length of all the recordings in the dataset. In this work, there is not such a straightforward choice for a sequence length, as all Xeno-canto recordings have variable lengths. The designer of SEDnet[31] was faced with the same issue and chose for a sequence length of 512 frames in SEDnet's publicly available implementation[9]. This sequence length would cover approximately 3 seconds of audio, given the audio sampling rate of 44.1kHz and the STFT window length of 512 used in this project (see Equation 1 for the conversion formula). Such a short time frame would make results hard to interpret, give the model little temporal context, and slice many bird songs into segments, which would defeat the purpose of using a PSED model for finding song on- and offsets. For this work, we chose a sequence length of 2048 frames, which coveres approximately 11 seconds of audio. A sequence length of 2048 was the largest multiple of 512 that still allowed for a batch size of 30 during training.

The code supplied with this work contains a script that automates the creation of features given a set of audio files and annotation files. It allows users to set different spectrogram window lengths, sequence lengths, audio sample rates, and more. This should allow for easy parameter experimentation in future work.

[8]https://github.com/dynilib/dynitag

[9]https://github.com/sharathadavanne/sed-crnn

Figure 3 shows a visual summary of all the above-mentioned steps in the dataset creation pipeline. Green blocks represents programs and arrows represent flows of data. Table I shows the number of annotations that were made for each bird class.

Species	Annotations
Chiffchaff	883
Great tit	796
Blackbird	566
Warbler	503
Wren	500
Total	3248

TABLE I: Dataset statistics

IV. PSED Model

A. Implementation

This work features a complete rewrite of the publicly available code of SEDnet[9]. The code was rewritten for several reasons:

- To incorporate more modularity, extensibility and documentation to provide for future growth and code reuse.
- To reorganise the flow of tensors through the model to make the tensor operations more comprehensible.
- To use a data generator instead of variables during training and testing to accommodate datasets whose size exceeds the system's memory size.
- To log as much training information as possible to Tensorboard, instead of including plot generation code in the training loop.

The code was mainly written in Python3[10] and Keras[11], an open-source neural-network library. All code is fully compliant with the PEP8 style guidelines[12]. The shell scripts that automate the production of features from audio files and annotations were written for both Bash[13] and Fish[14].

B. Model parameters

Preliminary experiments on the TUT Sound events 2017 development dataset [34] showed that an increase in the number of features in every layer of the network significantly improved model performance. The number of features in the recurrent layers was multiplied by 8. The number of features in the convolutional layers was set to increase from 64 to 256, instead of using 128 features throughout all layers (see Table II for the this work's model architecture and the SEDnet codebase[9] for the original feature counts). Dropout was removed as it only weakened performance on bird PSED task. The Adam optimiser [36] was used with a learning rate of 0.001. Model loss was calculated using a binary crossentropy loss. Separate models were trained and tested on single, double and triple concurrent sound sources. Whenever technically feasible, models were trained 3 times with different cross validation seeds for 40 epochs with a batch size of 30. Exceptions were made for the model trained on 3 concurrent sound events, as the model's memory footprint exceeded system memory (see Table III). The data was apportioned into a training and a test set with a 80%-20% split.

[10]https://www.python.org/
[11]https://keras.io/
[12]https://www.python.org/dev/peps/pep-0008/
[13]https://www.gnu.org/software/bash/
[14]https://fishshell.com/

C. Data Generators

A novel addition to the SEDnet codebase was the data generator. A Keras generator was used to get batches of spectrograms and label arrays during training. The generator also greatly simplified the creation of mixed spectrograms and label arrays. During training, the generator selects a bird class at random and then selects a random spectrogram and label array pair from that class. Each sample is popped off a periodically refreshed list, to ensure that each sample has an equal probability of being selected and that no sample is selected a disproportionate number of times. Whenever concurrent sound sources are required, the generator merges spectrograms and label arrays.

Layer type	**Block occurences**	**Number of Features**
2D Convolution (3x3) Batch Normalization Relu Activation Max Pooling (2x1)	5x	64, 64, 128, 128, 256
Bidirectional Gated Recurrent Unit	2x	256, 256
Time Distributed Fully Connected	2x	256, 256
Sigmoid Activation	1x	

TABLE II: Network block & feature counts

V. Materials

All programs were run on an Ubuntu 16.04 desktop. All 3rd party Python3 modules that were used are listed in a requirements file in the codebase. Whenever a model needed to be trained, six GTX 1080 GPUs were rented on Vast.ai to accommodate for the model's large memory footprint. Vast.ai[15] is a cloud GPU rental market that allows users to rent and list machines. Adobe Photoshop[16] was used for editing some of the images that appear in this report. This document was prepared in Latex[17].

VI. Performance Metrics

The PSED model in this work is evaluated by comparing its predictions to ground truth label arrays. We use the F-score and error rate described in [24] as metrics, as they have been used in numerous PSED publications and have been the primary metrics used in the DCASE PSED challenges since 2016 [18].

[15]https://vast.ai/
[16]www.adobe.com/Photoshop
[17]https://www.latex-project.org/

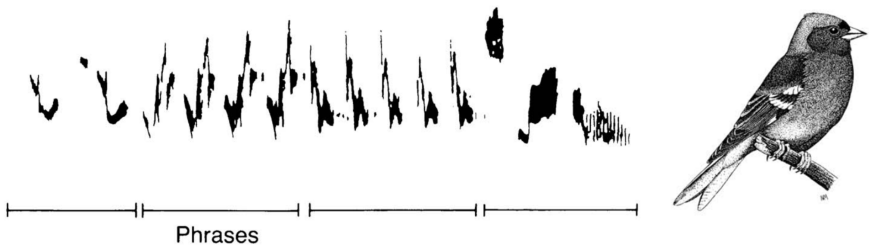


Fig. 4: Bird song phrases, image from [3]

The F-score considers both precision P and recall R, which in turn consider true positives TP, false positives FP and false negatives FN. The F-score is calculated as follows:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F = \frac{2 \cdot P \cdot R}{P + R}$$

The error rate ER measures the amount of errors in terms of insertions I, deletions D and substitutions S in N segments and is calculated as follows:

$$ER = \frac{\sum_{k=1}^{K} S(k) + \sum_{k=1}^{K} D(k) + \sum_{k=1}^{K} I(k)}{\sum_{k=1}^{K} N(k)}$$

$$S(k) = \min(FN(k), FP(k))$$

$$D(k) = \max(0, FN(k) - FP(k))$$

$$I(k) = \max(0, FP(k) - FN(k))$$

There are two ways one can count the intermediate statistics TP, FP, FN: by segment or by event. Segment-based counting entails that every spectrogram segment is treated as a multi-class classification problem. Event-based counting considers a series of segments as a whole, that is, whether or not a prediction event overlaps with a ground truth event. In this work the event-based counting method is not used for two reasons.

1) The PSED model occasionally predicts many short events (see Fig 11).
2) Single bird songs are sometimes annotated as many short events.

Given these two facts, the overlap between the predicted events and ground truth events would give a skewed view of model performance. Thus, this work will use frame-wise F-score and error rate as primary metrics. Note that segment-based metrics are called frame-wise in this work, as a segment of a spectrogram is called a frame and the term 'frame-wise' has been popularised by [31]. For visual examples of the above-mentioned metrics, see Fig 9 and Fig 10.

A Python implementation of these metrics was made publicly available[18] by the authors of [24]. This implementation was used throughout this work. To enable Keras to display the metrics in Tensorboard during training (see Fig 12), the code for the F-score and error rate was also rewritten in Keras variables and operators.

[18]http://tut-arg.github.io/sed_eval/

VII. Results

The average performance of the 3 final models can be found in Table III. The F-score and error rate are averaged over three cross-validation iterations for each model. For a minor qualitative analysis, one can refer to the 3 randomly chosen predictions in Fig 14 for each of the three models. The performance is surprisingly high for such a challenging task, considering the original SEDnet architecture [31] achieved an F-score of 71.7 and error rate of 0.43 on the TUT-SED 2009 dataset in 2017 [37]. As seen in Fig III, the inclusion of additional sound sources during training and validation only slightly reduced performance. This shows that the model was nearly as successful at detecting polyphonic sound events as it was at detecting isolated sound events.

We suspect that there are several factors that contributed to the model's high performance:

1) Some bird songs contain extremely repetitive structures. E.g. a single recording of a chiffchaff can contain many nearly identical phrases (see Fig 13). This means that the network is exposed to many similar cues that can all reinforce the same prediction.
2) Due to the batch size of 30, the 11 second sequence length and the high temporal resolution, the model is exposed to more than five minutes of high resolution data for every single weight update. This may have contributed to the model's stable learning progression (see Fig 12). This effect is multiplied by the number of species in a sample for the models trained on multiple audio sources.
3) The data used for annotations was exceptionally clean. Only A class recordings were considered, which are often recorded by Xeno-canto users with professional gear, such as parabolic microphones. All annotations were made by a single person, which promises some level of annotation consistency. To avoid mislabeling during the annotation process, recordings were skipped if the class of the bird songs was ambiguous. Other SED datasets, such as earlier mentioned TUT-SED 2009, feature more cluttered recordings and a larger variance in audio quality [38].
4) Even though there is absolutely no overlap in the training and test set, the data distributions may be very similar. That is to say, many of the recordings may share acoustic

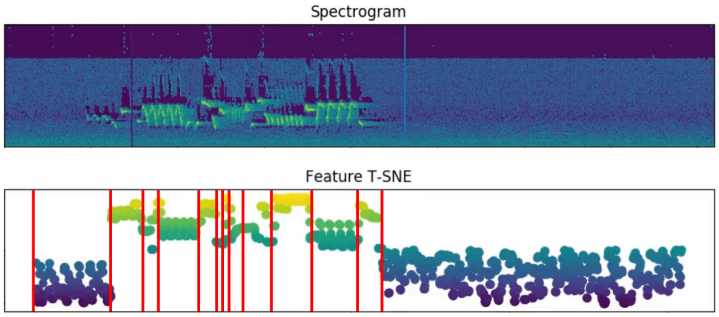


Fig. 5: t-SNE of CNN features covarying with phrases

features as they may have been recorded by the same person with the same gear. The Xeno-canto recording contribution graphs[19] follows a power law distribution, which means that a large portion of the repository has been contributed by a small group of members.

5) The spectrogram window length of 512 is relatively short when compared to window lengths used in other works on bird sound recognition [16] [12]. Upon manual inspection, this high temporal resolution reveals structures in the spectrograms that are distinct in outline and detail (see Fig 8). This sharpness may benefit the network's pattern detection capabilties.

Audio Sources	Batch size	Epochs	Avg. F-score	Avg. error rate
1	30	40	0.97	0.05
2	30	40	0.95	0.10
3	22	40	0.94	0.11

TABLE III: Results

A. Song Phrases

Surprisingly, the trained models were found to have the capacity to segment bird song phrases. A song phrase is defined as a series of units which occur together in a particular pattern [3]. Without going into the terminology of song units, one can refer to Fig 4 for a visual example of bird song phrases. The discovery was made during model debugging. A one dimensional t-SNE [39] plot was used to gain insight into the model's inner representation of the data (see Fig 5). For each inference, the feature vectors from the final CNN layer were intercepted. A one dimensional t-SNE mapping of these feature vectors was then plotted along a time axis, such that the visualisation would align with the spectrograms and label arrays. As can be seen in Fig 5, a switch in phrases in the spectrogram is accompanied by a change of the t-SNE mapping. To test whether these phrases could be automatically segmented, a K-means clustering algorithm was used on unmodified CNN feature vectors. The number of cluster centroids was manually tuned for demonstration purposes, but could be automatically determined by, for instance, using the elbow method[20]. Each red line in Fig 5 indicates the switch between the most dominant cluster in the local neighbourhood of 3 time steps.

While this segmentation approach appears to be promising, no performance metrics were calculated in this work as such an evaluation would require a dataset with phrase annotations.

VIII. Discussion

While building the dataset and PSED model for this work, many questions regarding future work arose. We would like to consider how the model and dataset could be improved, what steps could be taken to scale them up, and how they could be modified to distill other types of information from audio.

Annotating the dataset was the most time-consuming task in this work. Expanding the dataset to contain enough species for effective biomonitoring will either require a collaborative effort or some fundamental alterations to the model's architecture. The authors of [37] recently showed that it is possible to teach a network to perform PSED by using only weakly labeled data. They achieved this by adding an auxiliary classification branch after the time distributed fully connected layer and training the model on a classification task. The weak labels would allow the network to learn features for classification, which could thereafter be used for PSED. Such a model may be very powerful when combined with the hundreds of thousands of recordings on Xeno-canto and a rigorous hyperparameter and architecture search.

Another interesting finding, published in [40] by the same team, was that the model can be extended to localise and track sound sources. They accomplished this by adding a regression branch that predicts a sound's direction of arrival over time. For bird monitoring, one could generate a synthetic dataset by simulating moving bird sound sources and stereo microphones. Being able to track sources would greatly simplify and possibly solve the problem of counting individual birds in an audio scene.

As a final point of interest, we would like to consider the possible implications of the phrase segmentation method presented in this work. It could be used to replace or aid the very involved process that is currently used for creating

[19]https://www.Xeno-canto.org/collection/stats/recordists

[20]https://en.wikipedia.org/wiki/Elbow$_m ethod_(clustering)$

phrase catalogues[21]. The idea of manipulating the network's inner representation of bird songs could also be taken a step further. One may use it to distill other kinds of information given a different learning task. For instance, training a model to distinguish between birds of the same species may give rise to latent variables that are interpretable, such as a specimen's age, sex or fitness. A classifier trained to distinguish bird species may reveal features whose differences correspond to genetic distance. These and other possible implicit features would bypass the need for expensive labeled datasets and make neural networks even more valuable as biomonitoring tools.

IX. Conclusion

In this work a dataset was created that covers more than 3200 bird sounds. The dataset was used to train a polyphonic sound event detection neural net model that was able to recognise up to three overlapping sound events with an F-score of 0.94. This performance is well above that of similar PSED models used in other domains, which can most likely be attributed to the data cleanliness and carefully selected hyperparameters. The full journey from initial experiments to dataset assemblage and model training has been documented in this report and in the publicly available codebase. We invite those that may wish to continue this research to contribute to the codebase and dataset. We also hope that this work will serve as a motivator for future research into the applications of neural networks in biomonitoring.

Acknowledgment

I wish to express my sincere thanks to Arnoud Visser. Without his guidance, weekly sit-down sessions and critical eye, this paper would not have materialised. I would also like to thank my second supervisor Ashley Burgoyne, the University of Amsterdam, all Xeno-canto contributors, my family and Lucia.

References

[1] J. Kamp, S. Oppel, H. Heldbjerg, T. Nyegaard, and P. F. Donald, "Unstructured citizen science data fail to detect long-term population declines of common birds in denmark," *Diversity and Distributions*, vol. 22, no. 10, pp. 1024–1035, 2016.

[2] A. Johnston, S. E. Newson, K. Risely, A. J. Musgrove, D. Massimino, S. R. Baillie, and J. W. Pearce-Higgins, "Species traits explain variation in detectability of uk birds," *Bird Study*, vol. 61, no. 3, pp. 340–350, 2014.

[3] C. K. Catchpole and P. J. Slater, *Bird song: biological themes and variations*. Cambridge university press, 2003.

[4] W. A. Searcy and M. D. Beecher, "Song as an aggressive signal in songbirds," *Animal Behaviour*, vol. 78, no. 6, pp. 1281–1292, 2009.

[5] S. D. Hill, D. H. Brunton, M. G. Anderson, and W. Ji, "Fighting talk: Complex song elicits more aggressive responses in a vocally complex songbird," *Ibis*, vol. 160, no. 2, pp. 257–268, 2018.

[6] J. M. Reid, P. Arcese, A. L. Cassidy, S. M. Hiebert, J. N. Smith, P. K. Stoddard, A. B. Marr, and L. F. Keller, "Fitness correlates of song repertoire size in free-living song sparrows (melospiza melodia)," *The American Naturalist*, vol. 165, no. 3, pp. 299–310, 2005.

[7] K. L. Buchanan, K. A. Spencer, A. Goldsmith, and C. Catchpole, "Song as an honest signal of past developmental stress in the european starling (sturnus vulgaris)," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 270, no. 1520, pp. 1149–1156, 2003.

[8] L. Z. Garamszegi, A. P. Møller, J. Török, G. Michl, P. Péczely, and M. Richard, "Immune challenge mediates vocal communication in a passerine bird: An experiment," *Behavioral Ecology*, vol. 15, no. 1, pp. 148–157, 2004.

[9] S. M. Redpath, B. M. Appleby, and S. J. Petty, "Do male hoots betray parasite loads in tawny owls?" *Journal of Avian Biology*, vol. 31, no. 4, pp. 457–462, 2000.

[10] J. Falls, J. Krebs, and P. McGregor, "Song matching in the great tit (parus major): The effect of similarity and familiarity," *Animal Behaviour*, vol. 30, no. 4, pp. 997–1009, 1982.

[11] M. D. Beecher, P. K. Stoddard, E. S. Campbell, and C. L. Horning, "Repertoire matching between neighbouring song sparrows," *Animal Behaviour*, vol. 51, no. 4, pp. 917–923, 1996.

[12] D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge," *Methods in Ecology and Evolution*, vol. 10, no. 3, pp. 368–380, 2019.

[13] D. Stowell, "Freefield1010-an open dataset for research on audio field recording archives," 2013.

[14] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. P. Bello, "Birdvox-full-night: A dataset and benchmark for avian flight call detection," in *Proc. IEEE ICASSP*, (Calgary, Canada), Apr. 2018.

[15] W.-P. Vellinga and R. Planqué, "The xeno-canto collection and its relation to sound recognition and classification.," in *CLEF (Working Notes)*, 2015.

[16] S. Kahl, T. Wilhelm-Stein, H. Klinck, D. Kowerko, and M. Eibl, "Recognizing birds from sound - the 2018 birdclef baseline system," *CoRR*, vol. abs/1804.07177, 2018. arXiv: 1804.07177. [Online]. Available: http://arxiv.org/abs/1804.07177.

[17] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, and J. P. Bello, *BirdVox-full-night: a dataset for avian flight call detection in continuous recordings*, Oct. 2017. DOI: 10.5281/zenodo.1205569. [Online]. Available: https://doi.org/10.5281/zenodo.1205569.

[21] https://marce10.github.io/2017/03/17/Creating$_s ong_c atalogs.html$

[18] G. Dekkers and P. Karsmakers, *DCASE 2018, Task 5: Monitoring of domestic activities based on multi-channel acoustics - Development dataset*, May 2018. DOI: 10.5281/zenodo.1247102. [Online]. Available: https://doi.org/10.5281/zenodo.1247102.

[19] A. Joly, H. Goëau, C. Botella, H. Glotin, P. Bonnet, W.-P. Vellinga, R. Planqué, and H. Müller, "Overview of lifeclef 2018: A large-scale evaluation of species identification and recommendation algorithms in the era of ai," in *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, 2018, pp. 247–266.

[20] T. Grill and J. Schlüter, "Two convolutional neural networks for bird detection in audio signals," in *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE, 2017, pp. 1764–1768.

[21] H. Goëau, H. Glotin, R. Planqué, W.-P. Vellinga, S. Kahl, and A. Joly, "Overview of birdclef 2018: Monophone vs. soundscape bird identification," *CLEF working notes*, 2018.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[23] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[24] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[25] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen, "Sound event detection in multisource environments using source separation," in *Machine Listening in Multisource Environments*, 2011.

[26] J. Dennis, H. D. Tran, and E. S. Chng, "Overlapping sound event recognition using local spectrogram features and the generalised hough transform," *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1085–1093, 2013.

[27] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *2010 18th European Signal Processing Conference*, IEEE, 2010, pp. 1267–1271.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[29] G. Parascandolo, H. Huttunen, and T. Virtanen, "Recurrent neural networks for polyphonic sound event detection in real life recordings," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 6440–6444.

[30] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 international joint conference on neural networks (IJCNN)*, IEEE, 2015, pp. 1–7.

[31] S. Adavanne, P. Pertilä, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," *CoRR*, vol. abs/1706.02291, 2017. arXiv: 1706.02291. [Online]. Available: http://arxiv.org/abs/1706.02291.

[32] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[33] S. Round. (2019), [Online]. Available: https://www.vogelvisie.nl/top50.php.

[34] A. Mesaros, T. Heittola, and T. Virtanen, *Tut sound events 2017, development dataset*, Mar. 2017. DOI: 10.5281/zenodo.814831. [Online]. Available: https://doi.org/10.5281/zenodo.814831.

[35] M. Lasseck, "Audio-based bird species identification with deep convolutional neural networks.," in *CLEF (Working Notes)*, 2018.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] S. Adavanne, P. Pertilä, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017, pp. 771–775.

[38] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *2010 18th European Signal Processing Conference*, IEEE, 2010, pp. 1272–1276.

[39] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[40] S. Adavanne, A. Politis, and T. Virtanen, "Localization, detection and tracking of multiple moving sound sources with a convolutional recurrent neural network," *arXiv preprint arXiv:1904.12769*, 2019.

APPENDIX

$$\text{Frames in 1 second} = \frac{\text{sample frequency}}{\text{window length}/2} \qquad (1)$$

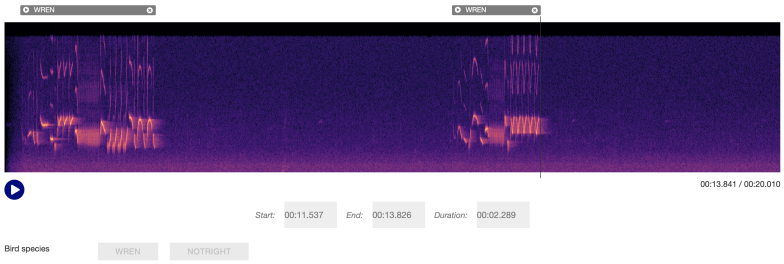


Fig. 6: Annotation interface

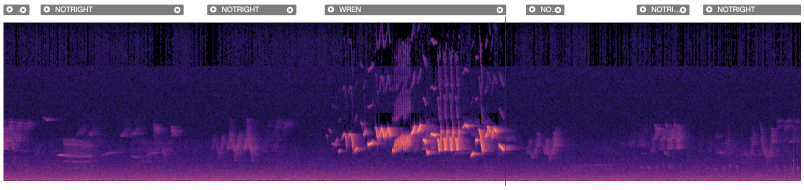
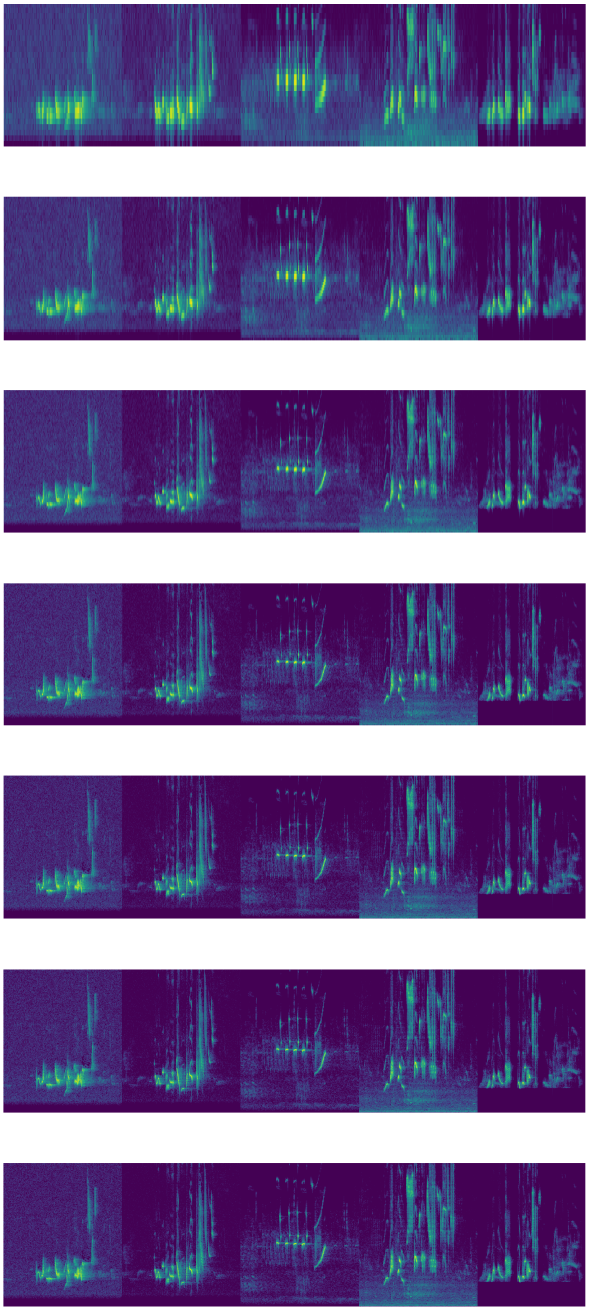


Fig. 7: Suppressing unwanted sounds



copy.png

Fig. 8: Spectrograms for different STFT window lengths

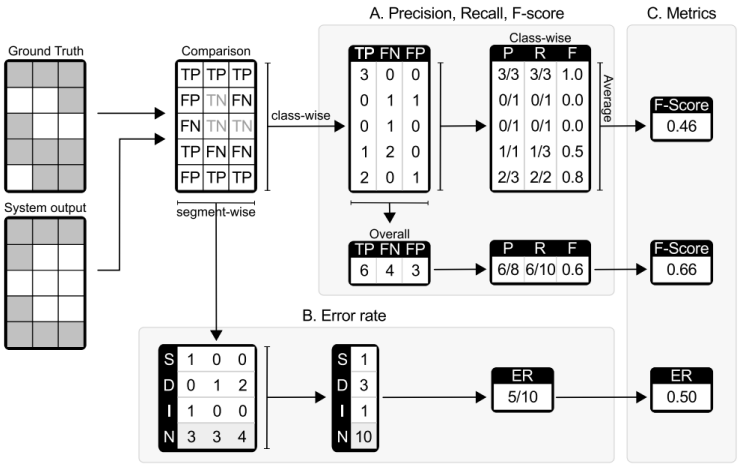


Fig. 9: Segment-based metrics for SED

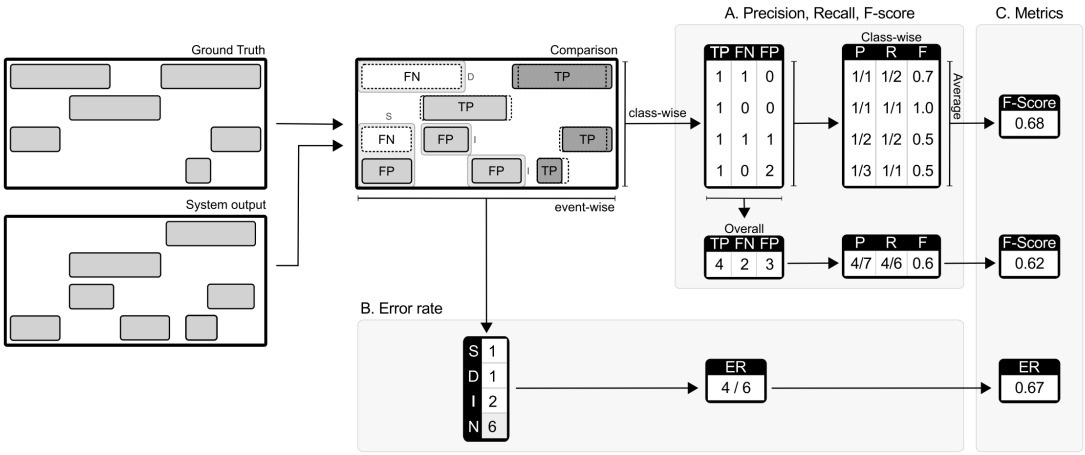


Fig. 10: Event-based Metrics for SED

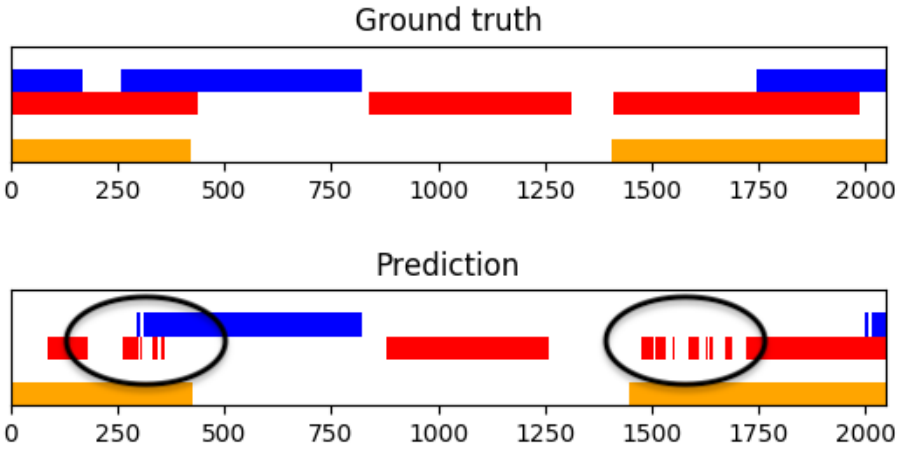


Fig. 11: Rapid Bursts of unconfident predictions

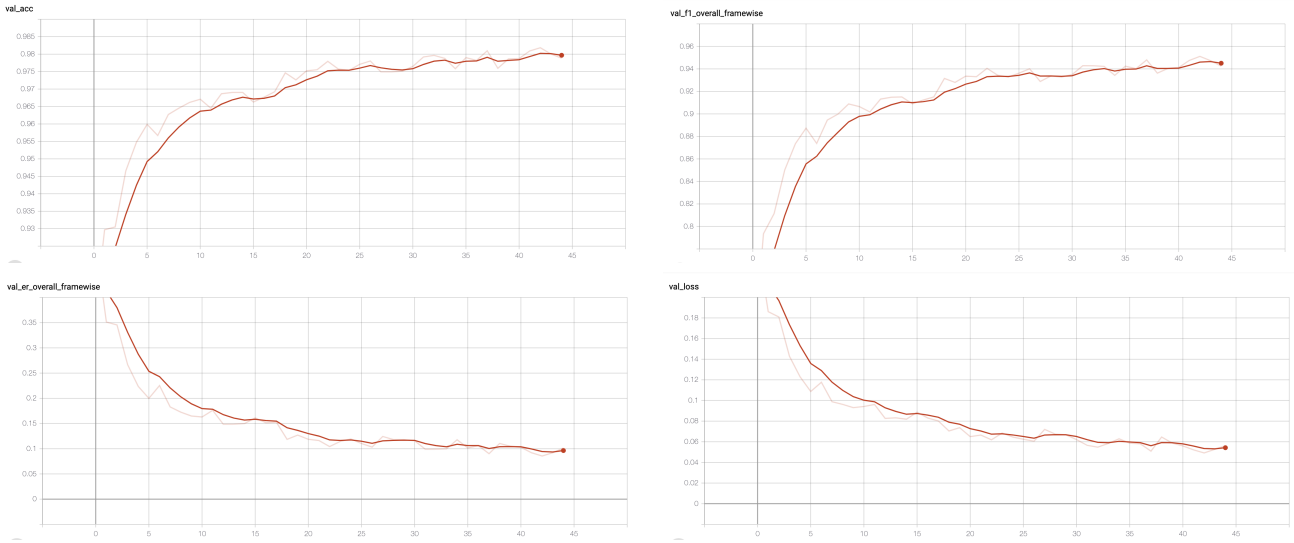


Fig. 12: Accuracy, F-score, error rate and loss on a validation set during training for 3 concurrent sound sources

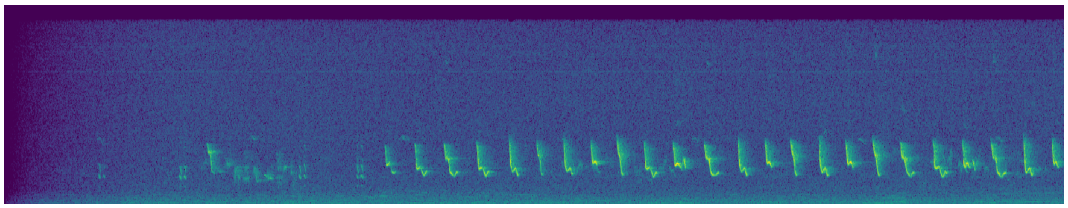


Fig. 13: Spectrogram of a chiffchaff's song

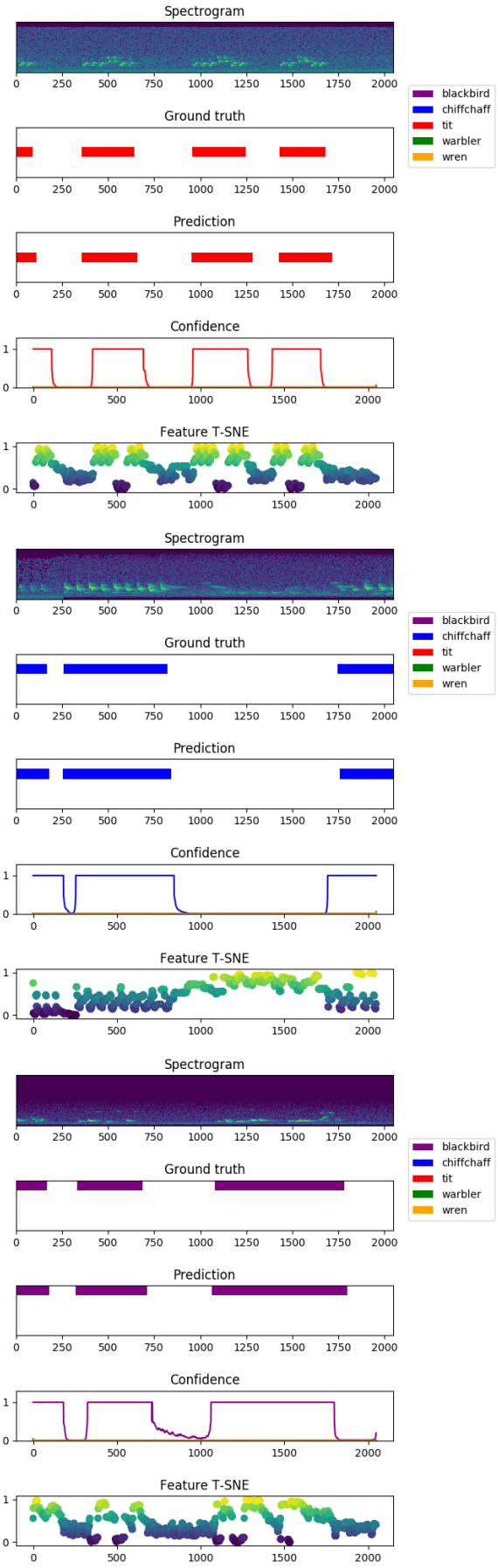


Fig. 14: A sample of predictions for the single source model

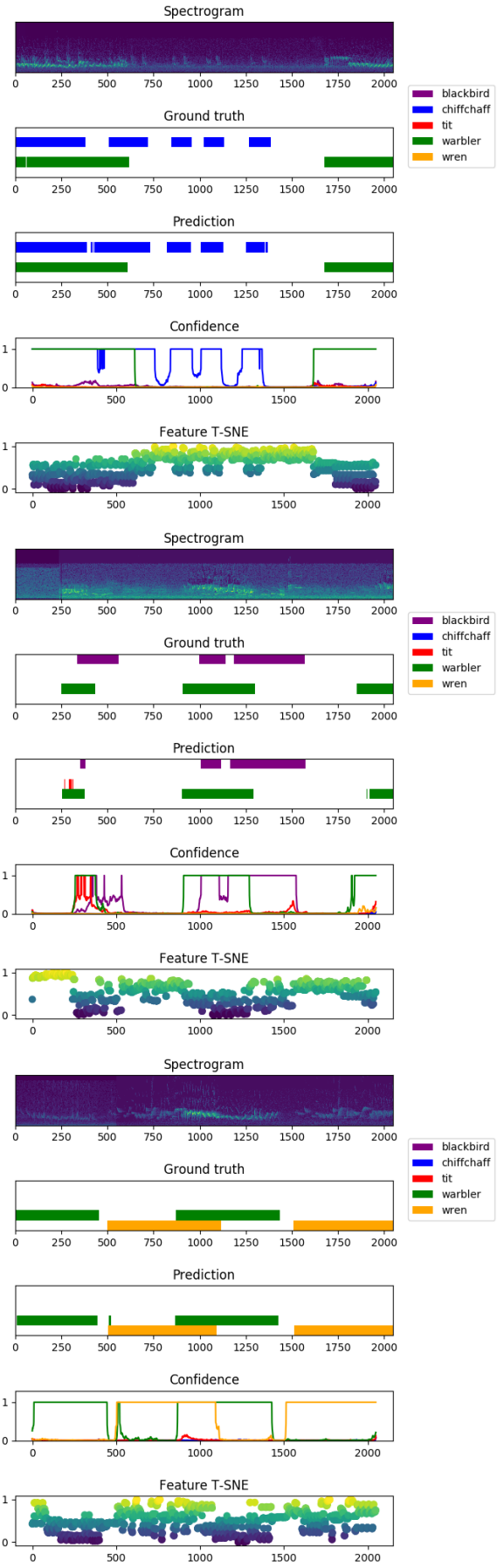


Fig. 15: A sample of predictions for the 2 concurrent sources model

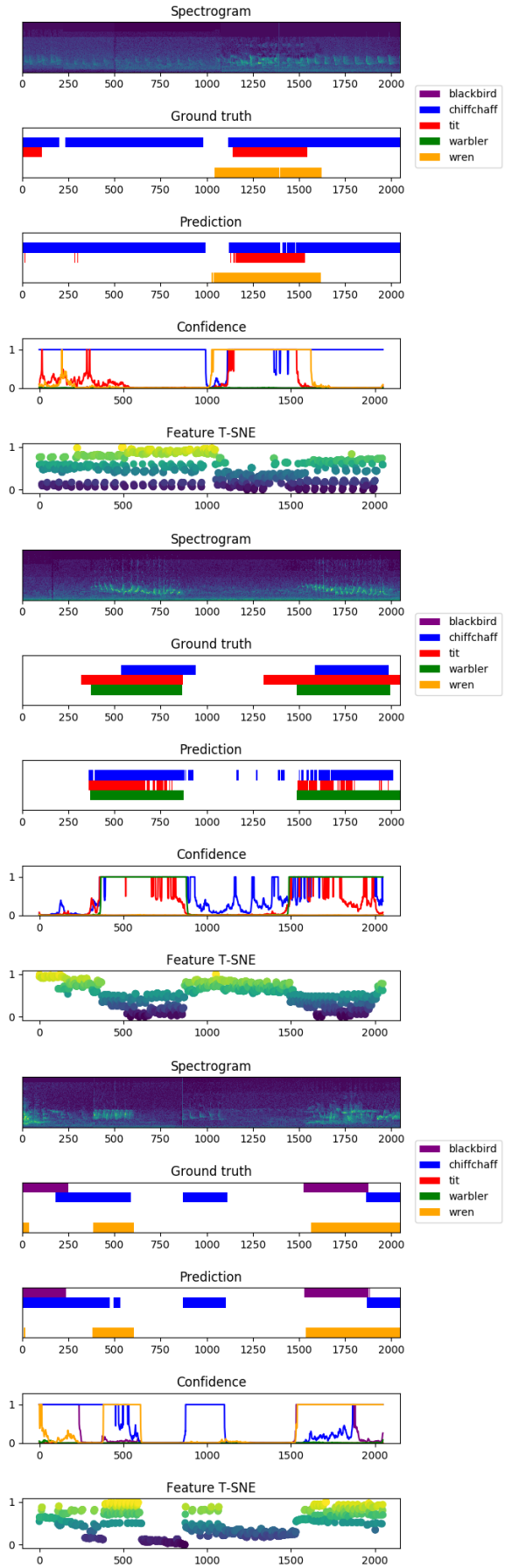


Fig. 16: A sample of predictions for the 3 concurrent sources model