Hear-and-avoid for unmanned air vehicles using convolutional neural networks

Wijnker, D.C.; van Dijk, Tom; Snellen, M.; de Croon, G.C.H.E.; de Wagter, C.

DOI
10.1177/1756829321992137

Publication date
2021

Document Version
Final published version

Published in
International Journal of Micro Air Vehicles

International Journal of Micro Air Vehicles

Original Research

International Journal of Micro Air Vehicles
Volume 13: 1–15

DOI: 10.1177/1756829321992137
journals.sagepub.com/home/mav
SAGE

Hear-and-avoid for unmanned air vehicles using convolutional neural networks

Dirk Wijnker[1], Tom van Dijk[1], Mirjam Snellen[2], Guido de Croon[1] and Christophe De Wagter[1]

Abstract

To investigate how an unmanned air vehicle can detect manned aircraft with a single microphone, an audio data set is created in which unmanned air vehicle ego-sound and recorded aircraft sound are mixed together. A convolutional neural network is used to perform air traffic detection. Due to restrictions on flying unmanned air vehicles close to aircraft, the data set has to be artificially produced, so the unmanned air vehicle sound is captured separately from the aircraft sound. They are then mixed with unmanned air vehicle recordings, during which labels are given indicating whether the mixed recording contains aircraft audio or not. The model is a convolutional neural network that uses the features Mel frequency cepstral coefficient, spectrogram or Mel spectrogram as input. For each feature, the effect of unmanned air vehicle/aircraft amplitude ratio, the type of labeling, the window length and the addition of third party aircraft sound database recordings are explored. The results show that the best performance is achieved using the Mel spectrogram feature. The performance increases when the unmanned air vehicle/aircraft amplitude ratio is decreased, when the time window is increased or when the data set is extended with aircraft audio recordings from a third party sound database. Although the currently presented approach has a number of false positives and false negatives that is still too high for real-world application, this study indicates multiple paths forward that can lead to an interesting performance. Finally, the data set is provided as open access.

Keywords

Unmanned air vehicle, collision avoidance, hear and avoid, convolutional neural network, aircraft detection

Date received: 14 February 2020; accepted: 12 November 2020

Introduction

More and more UAVs are entering the air every day, both for professional as well as for recreational purposes. Safety and regulations are subjects undergoing intense study nowadays in the UAV industry, as UAVs form a hazard for people, other (air) traffic, buildings, etc. For this research, the focus is on the collisions between UAV and manned air traffic. For example, emergency helicopters sometimes fly low in UAV-permitted airspace. Part of this problem can be solved by establishing flight rules, but a backup solution based on technology is desired. Technology becomes even more important when UAVs have to operate fully autonomously, as required by many future applications. A project initiated by Single European Sky ATM Research (SESAR) that aims to increase air traffic safety regarding to UAVs is called percevite.[a] Using multiple lightweight, energy-efficient sensors, collisions should be avoided in order to protect UAVs and their environment. One such a sensor is a microphone, which fulfills the task of 'hear-and-avoid', meaning that it should detect and avoid air traffic by sound. The goal of this research is to create a safer airspace by creating this hear-and-avoid algorithm.

[1]MAVLab, TUDelft, Delft, the Netherlands
[2]Aircraft Noise and Climate Effects, TUDelft, Delft, the Netherlands

Corresponding author:
Christophe De Wagter, Micro Air Vehicle Lab, Delft University of Technology, Kluyverweg 1, Delft 2629HS, the Netherlands.
Email: c.dewagter@tudelft.nl

The first feasibility study for hear-and-avoid has been performed by Tijs et al.[1] In that research, an acoustic vector sensor is used to detect other flying sound sources. de Bree and de Croon[2] also used an acoustic vector sensor in order to detect sound onboard UAV for military purposes. However, neither has used deep artificial neural networks to separate aircraft and UAV sounds. Other research groups have tried to identify the position of other UAVs using sound recorded from a UAV. Basiri and Schill, and Basiri et al.[3–6] tried to determine the position of a UAV in a swarm of UAVs. The transmitting UAV sends a chirp sound in the air that has frequencies different than the UAV's ego-sound, which can be picked up quite well while flying. They also do tests with engines of the receiving UAV turned off and the transmitting UAVs not transmitting the chirp anymore. Here too, the engine sounds of the transmitting UAV can help localize the platform. The 'hear-and-avoid' algorithm can be seen as a follow-up of these researches, and will address the identification of other air traffic by its original sound while also having the motors of the UAV producing sound. Harvey and O'Young[7] show that with two microphones, the detection of another UAV can be performed at such a distance that is double the distance needed to prevent a head-on collision. Furthermore, research is performed focusing only on the UAV sound by Marmaroli et al.[8] They have created an algorithm that is able to denoise the ego-sound of the UAV based on the knowledge about the propeller revolutions per minute (RPM).

One of the reasons that there is not a large amount of research performed on audio analysis for UAVs is that there are alternatives that provide traffic information, such as ADS-B, GPS, vision, etc. However, all alternatives have their disadvantages and do not fully eliminate the chance of a collision. For example, ADS-B requires a system in an aircraft that is not always present or turned on. For vision-based sense-and-avoid, its images can be disturbed due to speed, rain, fog, darkness, objects, etc. Sound, on the other hand, is inevitable for motorized aircraft, so it is a promising method. Moreover, microphones are lightweight, easy to use, omnidirectional and only weakly influenced by weather. The challenge that sound brings in this application is that many different sounds are present, such as the UAV ego-noise, wind, air traffic, ground traffic sounds and environmental sounds.

In this research, the following situation is studied: a UAV, which is carrying a single microphone, flies around and should detect incoming or passing aircraft based on sound. The detection of aircraft will be realized by means of a convolutional neural network (CNN). The performance of CNN on sound classification was found to be promising in McLoughlin et al., Phan et al.[9,10] and Zhang et al.[11] Detections could be used to warn a human operator or to simply land when other (manned) air traffic is present. The required dataset, which consists of audio recordings taken on a UAV including aircraft sound, did not exist and was artificially created. It is made open-source.[12] The CNN uses three audio features as input: Mel frequency cepstral coefficients (MFCCs), spectrograms and Mel spectrograms. Four variables are changed in the data sets to discover their influence: the window length, the amplitude ratio UAV/aircraft, the type of labeling and the use of third party database recordings.

The remainder of the article is structured as follows. The generation of the data set is explained in Section Audio acquisition, including how the individual sound recordings are obtained, how those are processed and mixed to recordings that include both UAV and aircraft sound. Secondly, the features and the model are described in Section Aircraft audio event recognition. The results for each of the models are shown in Section Results and discussed in Section Discussion.

Audio acquisition

A database that contains audio recordings, recorded on UAVs, of the UAV ego-sound and closely approaching aircraft forms the basis of any learning system. Because it was not available, a database was created that consists of (preprocessed) sound recordings (of UAVs, aircraft and rotorcraft) and labels, which indicate whether only UAV sound is present or UAV and aircraft sound are present. The data set is provided open access.[12]

Sound recordings

The current Dutch regulations on UAV prevent the UAV to come in the vicinity of a flying manned aircraft. In order to still have a representative database of UAV sounds that include passing aircraft, the UAV sounds and aircraft sounds are recorded separately and mixed afterwards. Three types of recordings have been used: self-made recordings using a microphone on a UAV, general aviation aircraft recordings using a microphone array and aircraft recordings obtained from a third party audio database.

Recordings of the UAV sounds. The UAV sounds are recorded in the Cyberzoo indoor flight testing facility of the TU Delft. An 808 micro camera[b] is placed under a Parrot Bebop UAV, so that its body already blocks part of the UAV ego-sound. Between the UAV and the microphone, foam is used to absorb the mechanical vibrations. During the recordings, the UAV performed rotations and movements around its pitch, roll and yaw axes at different speeds. After recording, the data are

cropped to remove the silences at the beginning and at the end. These recordings are complemented with audio recordings from a mobile phone that filmed the UAV from a close distance. Effectively a total of 20 min of UAV recordings are used.

Recordings of general aviation flights. Since the most probable group to come in contact with UAVs is general aviation (GA) rotor- and aircraft, flyover data have been obtained at the biggest GA airfield of the Netherlands, Lelystad Airport, in collaboration with the Aircraft Noise and Climate Effects (ANCE) section of the TU Delft.

As Lelystad airport is expanding to a larger airfield, the runway is being extended, but the new part is not in use yet. This part of the runway is therefore a perfect place to obtain recordings as the aircraft would fly straight over the so-called "acoustic camera".

The acoustic camera, designed and built by the TU Delft[13] consists of an array with eight bundles of eight PUI AUDIO 665-POM-2735P-R microphones. The bundles are arranged in a spiral shape for optimal beamforming purposes. The microphones are covered in a foam layer to decrease the noise due to wind. Moreover, the array is covered in foam in order to absorb ground reflections. All the bundles are connected to a data acquisition box (DAQ) which samples the data at 50 kHz and sends it to the connected computer. Not only the DAQ is connected to the computer, but also an ADS-B receiver in order to receive aircraft position information. However, the ADS-B did not produce useful information as none of the GA aircraft broadcast ADS-B information. Moreover, a mobile phone camera is placed in the center of the array to capture the flyover on video, but these data are not used for this research. The setup of the acoustic camera is shown in Figure 1.

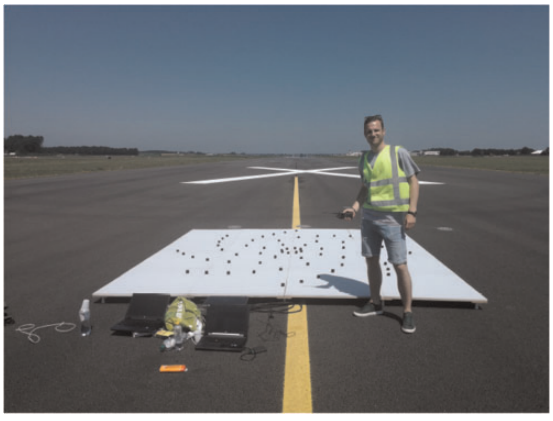

Figure 1. The acoustic camera on the runway of Lelystad Airport.

In total, 75 recordings are created, which consist of background noise recordings and flyovers. One recording sometimes consists of more than one flyover. Effectively, 75 GA aircraft and 9 helicopter flyovers are captured. The background noise consists of microphone noise, noise due to wind, distant traffic and a distant motor race track.

The data from every microphone are checked to make sure it worked correctly. One of the 64 microphones is faulty, so its data are not used. For this research, only the recording of one microphone is necessary.

Recordings obtained from a third party audio database. With regard to creating a data set that is representative for the possible air traffic sounds that a UAV could encounter, it had to consist of more than only flyover data. For example, other background noise could influence the detection performance. Therefore, also a (free) audio database[c] is consulted to obtain helicopter and (propeller) aircraft sounds. Only the sound samples that are of sufficient quality and which are not mixed with (too much) other background noise are selected manually.

Data preprocessing

All the separate recordings are manually checked before adding them together. For both the UAV recordings and the third party database aircraft recordings, only the in-flight part is kept. The recordings obtained at Lelystad airport are used entirely, and we manually labelled every second of data in the recording, indicating whether it consists of only background noise or includes aircraft sound. The recordings from Lelystad Airport include noise introduced by the microphones and the wind. A first-order Butterworth low-pass filter is used to remove the high-frequency noise. Most of the time, the aircraft sound information is in the frequency region lower than 100 Hz. Only during a flyover aircraft, sound information comes above this value. In order to capture the higher frequency content during a flyover but also to remove much of the wind noise during the rest of the time, the cut-off frequency is set on 2.5 kHz.

All the recordings are resampled to a sample rate of 8 kHz as there is no important information present above the Nyquist frequency of 4 kHz and it decreases the size of the data set significantly, which shortens the computational time. Secondly, the sound recordings are normalized by scaling the amplitude between -1 and 1, so that the amplitude of two recordings is similar. Before mixing aircraft and UAV sounds, data augmentation is applied to all the separate aircraft and UAV recordings in order to increase the size of the

data set. Three types of data augmentation are applied: addition of white noise, increase in pitch and decrease in pitch. The white noise is a randomly generated Gaussian distribution with mean 0 and a variance of 0.005. The pitch is increased and decreased by two semitones on the 12-tone. An increase of two semitones relates to $\sqrt[12/2]{2} \approx 1.12$ times the original frequency. After augmentation, the data set is four times its original size.

Mixing the recordings

In order to get sound samples that include both aircraft and UAV sound, the following mixing procedure is used. First, the whole data set is split up in a test set and in a training set. All the augmented versions of a sound sample are always in the same set as their original sound sample to ensure that the two sets are uncorrelated.

Secondly, each recording from Lelystad airport is combined with a randomly selected UAV recording of the same set. Mixing consists of adding a segment of the Lelystad airport sound sample, which has a random length, to one of the UAV recordings on a random starting position. The mixed sample therefore never consists of only aircraft sound. The total length of each mixed sample is equal to the length of the UAV recording, which is different for each recording.

Mixing the third party database recordings is done slightly differently than the method described for the Lelystad recordings because the third party database recordings contain aircraft sound 100% of the time. The difference between the two mixing methods is that not only a part of the recording is added to the UAV sound sample, but the whole recording is added instead (at a random starting position).

The detection model in this paper requires the inputs to be of equal length (more on this in Section Model). As this is not the case for the combined samples, the third step is to cut the combined samples to equal lengths. To maximize the amount of data in the sets, the cutting length is set to 51 , which is equal to the length of the shortest combined sound sample.

The amplitude ratio when mixing the UAV and aircraft sound is not always 1:1. In this work, four UAV/aircraft amplitude ratios will be used, namely 0:1 (which means no UAV sound), 1:1 (equal amplitudes), 1:4 (aircraft sound amplitude is four times larger) and 1:8 (aircraft sound amplitude is eight times larger). Most of the time, a ratio of 1:4 is used. This ratio is obtained as follows. Assuming the average sound pressure level (SPL) of a UAV at 1 m distance is 76 dB[d] and that of an aircraft at 300 m distance is 88 dB,[e] the difference between the SPLs of the two sounds is 12 dB. Equation (1) shows how the SPL is calculated from the pressure p_1 (which is the amplitude in the waveform) of a sound and a reference pressure p_0. Taking the amplitude of the UAV waveform as reference pressure and the aircraft waveform as p_1, an SPL of 12 is obtained when the aircraft waveform is four times larger. If the ratio 1:4 is corresponding to an airplane on 300 m distance, 1:1 corresponds to a distance of 1200 m and 1:8 to a distance of 150 m, following equation (2). In this equation, r_2 is the distance of interest, r_1 is the original distance, SPL_1 is the SPL at r_1 and SPL_2 is the SPL at r_2.

$$SPL = 20\log\frac{p_1}{p_0} \tag{1}$$

$$r_2 = r_1 \cdot 10^{\frac{|SPL_1 - SPL_2|}{20}} \tag{2}$$

Labels

Each second of a mixed sample is given a binary label, indicating whether there is other aircraft sound present (1) or not (0). The recordings from Lelystad airport are labeled manually before mixing. There are two types of labeling, called nearby detection labeling and distant detection labeling. Nearby detection labeling is partly based on listening to the sound, and partly on looking at the spectrogram. The spectrogram, which is shown in Figure 2 and elaborated on in Spectrogram, shows the amount of frequency content over time. Nearby detection labeling gives label 1 when a peak is visible in the spectrogram. By ear this is noticeable as more high frequency content is heard.

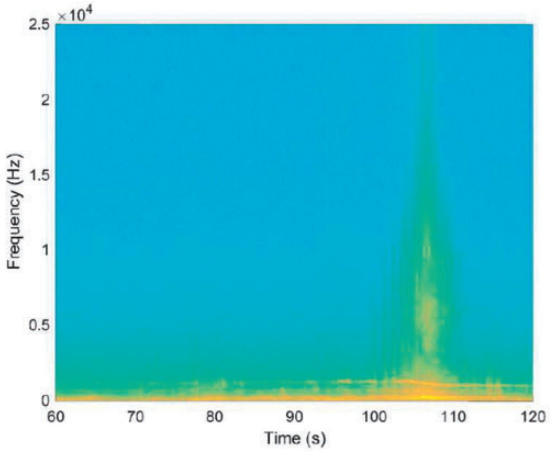

Figure 2. Spectrogram of a flyover recording. The exact flyover is between 100 and 110 s, which can be recognized by a yellow peak and a Doppler shift around 100 Hz. Also before and after the peak, the aircraft sound is present, which is visible by the horizontal line around 100 Hz.

Distant detection labeling is purely based on hearing. The frames in which a human is able to separate noise from aircraft sounds are labeled 1. This time it cannot be based on the spectrogram as the aircraft sound is either not visible on the spectrogram (when it is blended in too much with the background noise) or it is visible (as a line on a single frequency caused by the propeller's rotational speed) but the background noise is louder than the aircraft sound. An example of the latter is shown in Figure 3, at which the horizontal line around 100 Hz is also present when no label is given.

The time instances that are not labeled one are labeled zero, so also the background noise from the Lelystad recordings is given the same label as when there is no other aircraft sound present. In Figure 3, the areas in the spectrogram that are labeled as 1 are indicated in red for nearby detection labeling and green for distant detection labeling.

For the third party sound database, the whole aircraft recording is always labeled as a one, as each of the sound samples is selected on only having aircraft sounds. Again, all the time instances in the mixed recording that are not one are labeled zero.

Aircraft audio event recognition

The aircraft sound will be detected by a framework that exists of a feature extractor and a classifier. The features capture important sound information and reduce the dimensionality of the data. They are the inputs for the classifier. Thereafter, the classifier determines whether the sound sample contains aircraft sound or not.

Feature extraction

Three features are extracted from the combined sound samples using Python library Librosa.[14] First there are the Mel Frequency Cepstral Coefficients (MFCCs)[15] which are chosen because of their popularity in one of the biggest domains in machine hearing, automatic speech recognition (ASR). The two other features, the spectrogram and Mel spectrogram, are visual representations of the sound samples. Content-based analysis of images is already quite developed,[16] and therefore the image of a sound might be a good starting point.

For every feature, each frame in the time dimension has a length of 1 s. One second is a rather large frame but it chosen to reduce in dimensionality. The window moves over the sound sample with a step of 1 s. All the sound samples are 51 se long, and thus from each sound sample, 51 separate frames are obtained in the time dimension.

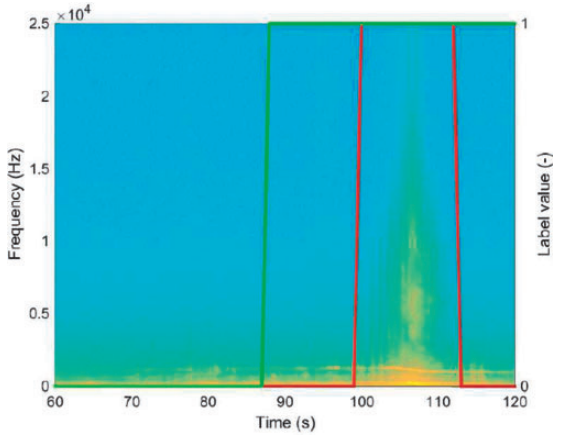

Figure 3. Spectrogram showing nearby detection labeling (red) and distant detection labeling (green).

MFCC

The cepstrum is a domain which represents the rate of change in multiple frequency bands. MFCCs are the coefficients of which the cepstrum is composed. It has the ability to separate convoluted signals in the time domain.[f] This domain is therefore often used in speech recognition, to separate the vocal pitch and the vocal tract. The coefficients are obtained by taking the logarithm of the amplitude spectrum, converting this to the Mel scale and taking the Discrete Cosine Transform (DCT). The Mel scale, which is expressed as a function of frequency (f) in equation (3), is a scale that approximates the human perception of frequency. This scale emphasizes the low frequencies (<1 kHz), which is also the frequency range in which most of the UAV/aircraft sound information is present. The full transformation from time domain signal to MFCC is shown in equation (4)[17]

$$M(f) = 2595\log\left(1 + \frac{f}{700}\right) \tag{3}$$

$$MFCC(d) = \sum_{k=1}^{K}(\log X_k)\cos\left[d\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right] \quad \text{for} \quad d = 0, 1, \ldots, D \tag{4}$$

where X_k is the discrete Fourier transform (DFT) obtained in equation (5) of which the frequency belonging to each k is warped to the Mel scale by equation (3). D is the total number of coefficients and N is the number of data point in the time frame. The number of coefficients used in this research is 20.

$$X_k = \sum_{n=0}^{N-1} X_n e^{-\frac{2\pi i}{N}kn} \quad \text{for} \quad k = 1, 2, \ldots, N \tag{5}$$

Spectrogram

Spectrograms are visual representations of the energy per frequency plotted against time, of which the Mel spectrogram uses the Mel scale of equation (3) on the frequency axis. A typical flyover spectrogram (without UAV sound) is shown in Figure 2. In this figure, the point where the aircraft is passing the array is between 100 and 110 s, which is visible with the large yellow peak and a Doppler shift (the sigmoid-shaped line around 1 kHz). It also shows that when the aircraft is further away, it lacks in high frequency content (due to atmospheric attenuation). That means most of the time only the aircraft's low frequency content is heard by the UAV in combination with low frequency noise.

The spectrograms are calculated following equation (6), which is the magnitude to the power p of the short-time Fourier transform (STFT). Usually the power spectral density (PSD) is chosen, for which $p=2$. It uses a window function $w[n]$, in this case the Hann window of 1 s, of which m is the index of the position in the window function with length N, discrete frequency k, signal $x[n]$ at time n

$$Spectrogram = \left| \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{\frac{-i2\pi kn}{N}} \right|^p \quad (6)$$

Model

The previously described features are the input for a deep artificial neural network: the convolutional neural network (CNN). It has shown best performance for sound event recognition tasks in McLoughlin et al., Phan et al.[9,10] and Zhang et al.[11] and.[11] The basic CNN used in this research is shown in Figure 4. The network is created with the Python libraries Keras[18] and Tensorflow.[19]

Even though the features consist of 51 s of UAV/aircraft sound, the input for the CNN is a smaller time window which slides over the time axis. The smaller time window is used as otherwise the detection output of a frame could be dependent on data from later frames, due to the fully connected layer. Multiple window lengths are used, as shown in Results. In the basis, however, the window size is 3 s. This window slides over the feature's time axis with a step of 1 s.

The first layers of the CNN are convolutional layers. There are two subsequent sets of layers, each consisting of two convolutional layers, followed by a max pooling layer. The convolutional layers use the rectified linear unit (ReLU) as activation function and it applies zero padding to the input. After the two sets, the output is flattened in order to be able to connect it with the output layer, a fully connected layer. For the output, a sigmoid activation function is used, which scales the output (as a float) between 0 and 1. The binary discrimination threshold determines whether this output becomes a 1 or a 0, so whether an aircraft is present or not, respectively. The network is based on Zhang et al.[11] and its parameters are modified based on preliminary test results.

Training the network is performed by means of a binary cross-entropy loss function and the Adam optimizer Kingma and Ba.[20] The Adam optimizer parameters are the same as in the original paper, so a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and no decay. After each pooling layer, dropout is used in order to prevent overfitting of the training data. The parameters for the CNN are shown in Table 1.

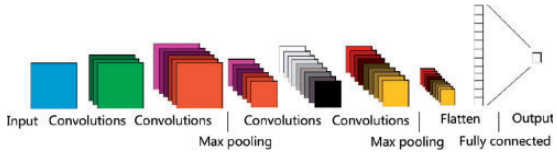

Figure 4. Architecture of the CNN. The input is a moving time window over the spectrogram, Mel spectrogram or MFCC. The output a binary value indicating whether aircraft sound is present or not.

Results

Each feature is combined with the CNN, so in total three models are tested. They are trained and tested on multiple data sets, which are listed in Table 2. To check the influence of certain parameters in the data set or in the model, four parameters are altered during the runs: the window length, the labeling type, the ratio in amplitude between the UAV and aircraft sound and whether third party database recordings and Lelystad airport recordings are used or only the Lelystad airport recordings.

There is one basis run, for which the window length is 3 s, the labeling is nearby detection labeling, the UAV/aircraft ratio is 1:4 and there are no third party

Table 1. Model parameters of the CNN from Figure 4.

Parameter	CNN
Convolution units first set	32
Convolution units second set	64
Kernel size	3×3
Pooling size	2×2
Dropout probability 1	0.25
Dropout probability 2	0.5

Table 2. Overview of the variables that are changed for each run, including their corresponding values and the values of the standard case, the basis run.

Variables	Basis values	Variations
UAV/Aircraft ratio	1:4	0:1 1:1 1:8
Third party database used	No	Yes
Labeling type	Nearby detection Labeling	Distant detection labeling
Window length (s)	3	10 15 20

Table 3. The number of each run with their corresponding changed variable and the corresponding value.

Run #	Variation
1	UAV/Aircraft ratio: 0:1
2	UAV/Aircraft ratio: 1:1
3	Basis run
4	UAV/Aircraft ratio: 1:8
5	Database used: Yes
6	Distant detection labeling
7	Window length: 10
8	Window length: 15
9	Window length: 20

database recordings involved. For all the other runs, only one variable of the basis run is changed each time.

The window length is either 3, 10, 15 or 20 s. The Lelystad airport recordings are labeled manually, in two manners, as explained in Section Labels. For distant detection labeling, the training is performed with distant detection labeling and the testing is performed with nearby detection labeling. The idea behind this method is that the model could learn aircraft sound when it is not so obviously present, so that detection when the aircraft is obviously present is outstanding. The amplitude ratio between the UAV and the aircraft is tested when no UAV sound is present, and for the ratios 1:1, 1:4 and 1:8. Lastly, the third party database sounds are either added to the data set or omitted.

From here on, each specific run is indicated by the number of the run given in Table 3. The performance of the models is compared for each of the variables (window length, label type, etc.). This comparison is based on the receiver operating characteristic (ROC) curve. The ROC curve shows the true positive rate (TPR) against the false positive rate (FPR) for all possible binary discrimination thresholds. The area under the curve (AUC) is a measure of accuracy of the binary classifier. In this research specifically, especially the region of low FPR is important, as it shows how many times the UAV would falsely decide to warn the operator or descend. For each point on the ROC curve, the desirable discrimination threshold can be extracted, which determines whether the output from the model is classified with label 1 or label 0.

Influence of the UAV/aircraft ratio

Runs 1, 2, 3 and 4 are simultaneously plotted for the CNNs in Figure 5. In general, the best performance is achieved for the cases where there is no UAV sound present (run 1). If the UAV's ego-sound is added to the aircraft sound with an amplitude ratio of 1:1 (run 2), the performance is the worst in all cases. The figures show that amplifying the aircraft sound increases performance; however, there is little increase between the ratio 1:4 and 1:8. The expected result is that the less UAV content is present, the more the performance would converge to the result of run 1. Only for the MFCC and Mel spectrogram, this trend is visible in the lower FPR region. Looking at the AUC, the MFCC and the spectrogram show no convergence to the ratio of 0:1. In the case of the Mel spectrogram, there is only a difference visible between the ratio of 1:1 and the others.

Influence of the third party database recordings

In the basis run, only the recordings from Lelystad airport are used. This means that all the recordings have (fairly) the same background noise and types of airplanes and they use the same recording equipment. In order to check how much the models rely on these characteristics, they are trained and tested with the third party database recordings as well for this run.

Figure 6 shows that for all the models, the addition of the third party database recordings improves the performance of the model. Only for the very low FPR (<0.01), the basis run performs better for the MFCC-CNN and the Mel spectrogram-CNN.

Influence of labeling

The third type of modification made in the data set relates to which labels are used for training. For all cases, the nearby detection labeling is used for testing. For training, however, one run uses distant detection labeling and one run uses nearby detection labeling. When an aircraft is approaching, the lower frequencies of its generated sound reach the ear first. This low frequency content is in the same range as the background noise. It is therefore expected that for distant detection labeling a better separation is found in the model between drone and aircraft and therefore would also better perform for the nearby cases. Figure 7, however, does not prove this hypothesis. This time, for all

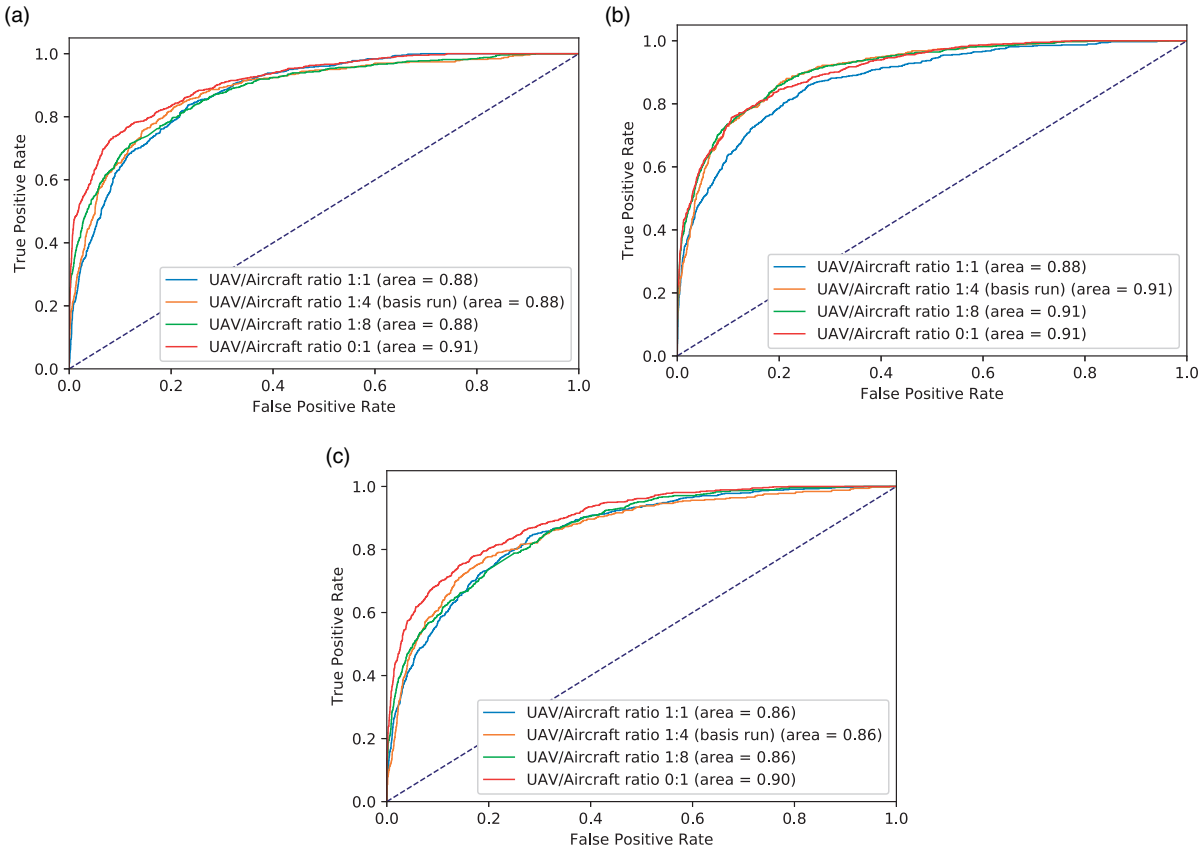

Figure 5. ROC curves showing the influence of the UAV/aircraft ratio for each feature. Best accuracy is achieved for the ratio 0:1 (no UAV sound present). The more UAV content is added, the worse the performance. (a) MFCC-CNN for different UAV/aircraft amplitude ratios. (b) Mel spectrogram-CNN for different UAV/aircraft amplitude ratios. (c) Spectrogram-CNN for different UAV/aircraft amplitude ratios.

features, the performance deteriorates when distant detection labeling is used.

Influence of the window length

The window length of the CNN determines how many seconds of history are used to determine whether the sound contains aircraft sound or only UAV sound. The more history the sound contains, the better the development of (possible) aircraft sound can be captured. It is therefore expected that with a larger window length, a better performance is achieved. However, eventually the performance of longer time windows are expected to converge as history from long ago does not give useful information in detecting aircraft sound in the present.

This hypothesis is confirmed for the CNNs using Mel spectrogram, spectrogram and MFCC in Figure 8. Improvement in AUC between a 3-s window and a 10-s window is shown in each of the subfigures. For window lengths of more than 10 s, the AUC hardly changes. For the spectrogram-CNN, there is a clear difference in the low FPR region between the 10 and 15 s.

Comparison of the features

So far, the results are only shown per feature. In order to show which feature works best, the features have been compared for the basis run in Figure 9. The results show that the Mel spectrogram performs best, followed by the MFCC. The spectrogram performs worst compared to the other two.

Even though the results are only set out for one run, this is true in general for the other runs. For the runs with a UAV/aircraft ratio of 0:1, 1:1, and distant detection labeling (run 1, 2 and 6), the MFCC is equally accurate as the Mel spectrogram. For the runs with an increase window size (run 7,8 and 9), the spectrogram is slightly better than the MFCC.

Moreover, a ROC curve with the binary discrimination threshold based on the pure energy of the signal is shown in Figure 9. This curve is used to see whether the model just checks the amount of energy in the signal or if it uses more elaborate features. The AUC gives away

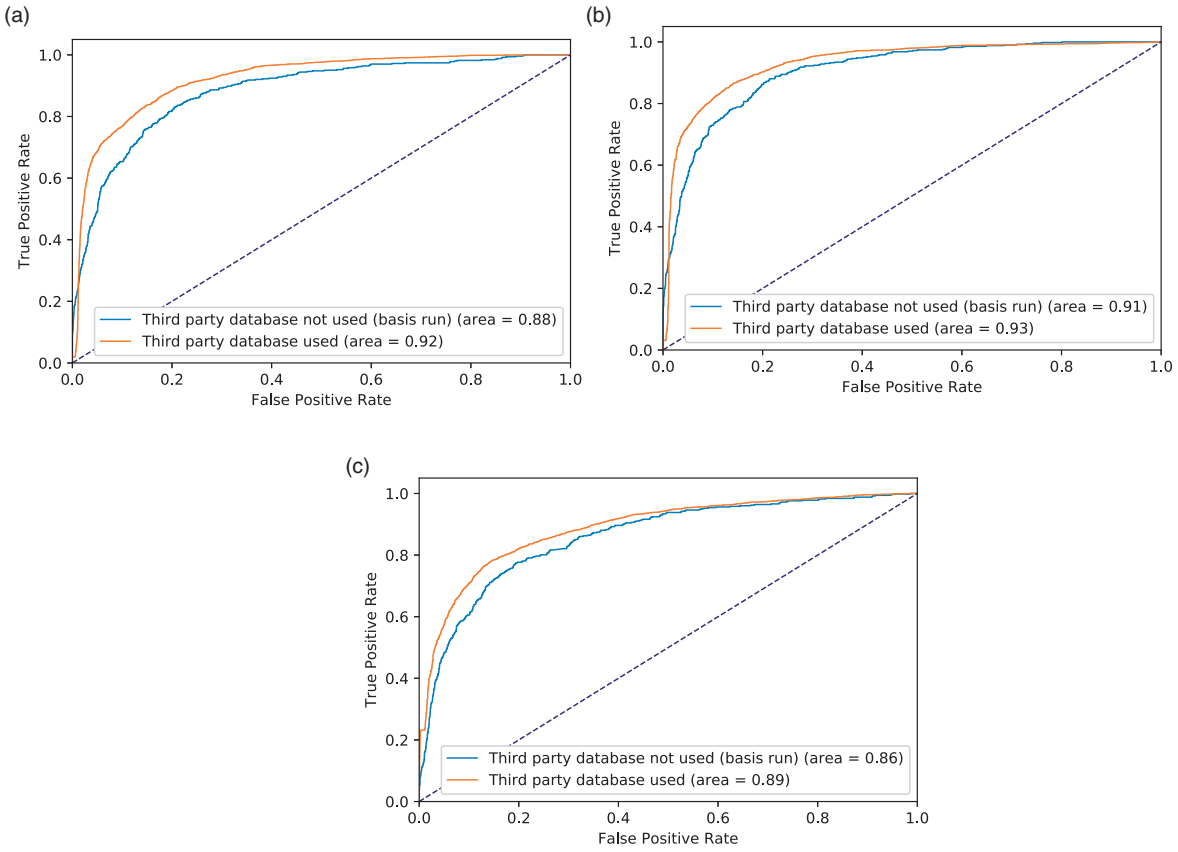

Figure 6. ROC curves showing the influence of the third party database recordings for each of the features. For all features, the performance increases using the third party database recordings. (a) MFCC-CNN with and without third party database recordings. (b) Mel spectrogram-CNN with and without third party database recordings. (c) Spectrogram-CNN with and without third party database recordings.

directly that the performance is significantly worse than the CNNs, so the model does not base its outputs simply on the amount of energy in the signal. Especially in the low FPR region (<0.1), the TPR is significantly lower than for the CNNs.

Visualization of the output

In order to clarify the output of the model, one of the runs is used to visualize the outputs. In Figure 10, the spectrogram of one sample of the basis run test set is shown, along with the expected label (in red), the output of the network (in black) and the binary discrimination threshold belonging to an FPR of 0.1 (in purple). This example shows a decent detection result in which the results in the time window for which the label is 1 (between 28 and 40 s) are correctly above the threshold (except for the first second). The rest of the output is always under the threshold and therefore not detected as an aircraft.

The correctness of the result of Figure 10, however, is not observed for all cases of the test set. False positives and false negatives are appearing as well, such as shown in Figure 11. In Figure 11, the time span between 30 and 45 s should be given a label of 1, but the model output is still under the threshold, except for 1 s. Also, the point at second 3 is just above the threshold, whereas it should be labeled 0. On the other hand, also for the human eye, the presence of an aircraft is better visible in the spectrogram of Figure 10 than in the spectrogram of Figure 11, due to the Doppler shift and the increase in energy (which can be seen by the increase of the yellow content) in Figure 10.

In order to confirm that the model can recognize closest point of approach (CPA) such as shown in the spectrogram, all the audio samples of the test set of the basis run are centered around the CPA (if any). For each second in the range of 10 s before the CPA and 10 s after the CPA, the mean values and standard deviation of the model output are taken. Those values are shown in Figure 12. Each dot represents the value of the mean, each bar the standard deviation from the mean. Figure 12 shows that at the CPA, the output

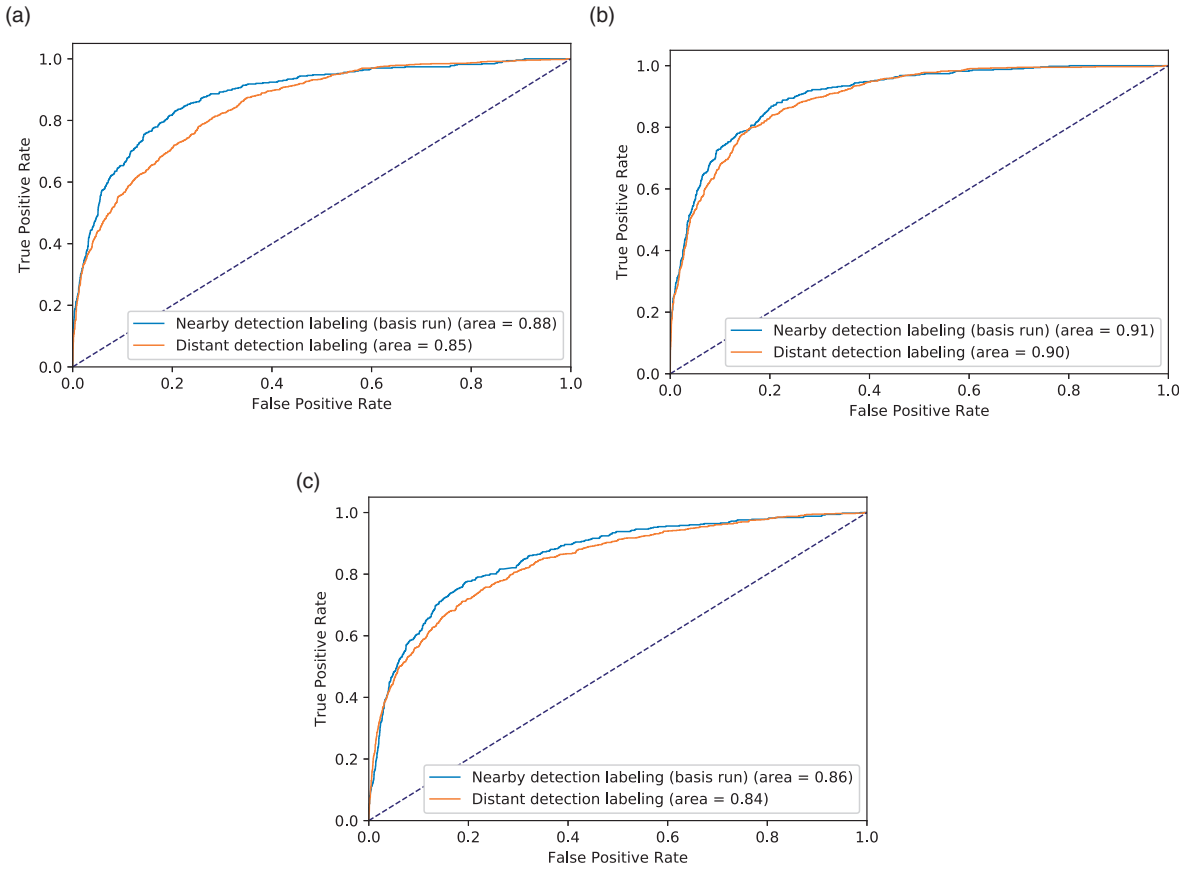

Figure 7. ROC curves showing the influence of labeling type for each of the feature. Each run is tested with nearby detection labeling. One run is using the nearby detection labeling for training as well and the other one uses the distant detection labeling during training. (a) MFCC-CNN comparing the performance for different label types. (b) Mel spectrogram-CNN comparing the performance for different label types. (c) Spectrogram-CNN comparing the performance for different label types.

value is usually the highest. Furthermore, the larger the time distance from the CPA, the lower the mean and standard deviation. There is, however, relatively much spread in the output of the network.

Precision and recall

The AUC gives a good overall indication for the accuracy of the model. However, in order to see how well the model performs per point on the ROC curve, precision and recall are used. Precision is defined in equation (7), in which FP is the number of false positives and TP is the number of true positives. For recall, also the false negatives FN are used, such as shown in equation (8)

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

In this research, an important value is $1 - recall$ for the label 0. This value shows how many false positives are present, so how often the UAV would falsely perform an avoidance maneuver. The recall for the label 1 is the second most important. It shows how well the aircraft is detected when it is present. The reason that it is less important than the $1 - recall$ for label 0 is because this value does not say when the false negatives appear. It is expected that the closer the aircraft gets, the better the detection performance. Figure 12 shows that this is actually the case for this model. So if the model does not detect the aircraft, it is probably not too close, so it would not directly lead to a critical situation. Precision shows how many of the predicted labels are relevant, which is less important for this application than the recall.

An example of the precision and recall and the confusion matrix for the Mel spectrogram-CNN with the window length 20 are shown in Tables 4 and 5, respectively. As a very low FPR beneficial, but still aircraft should detected, the point on the curve for which the

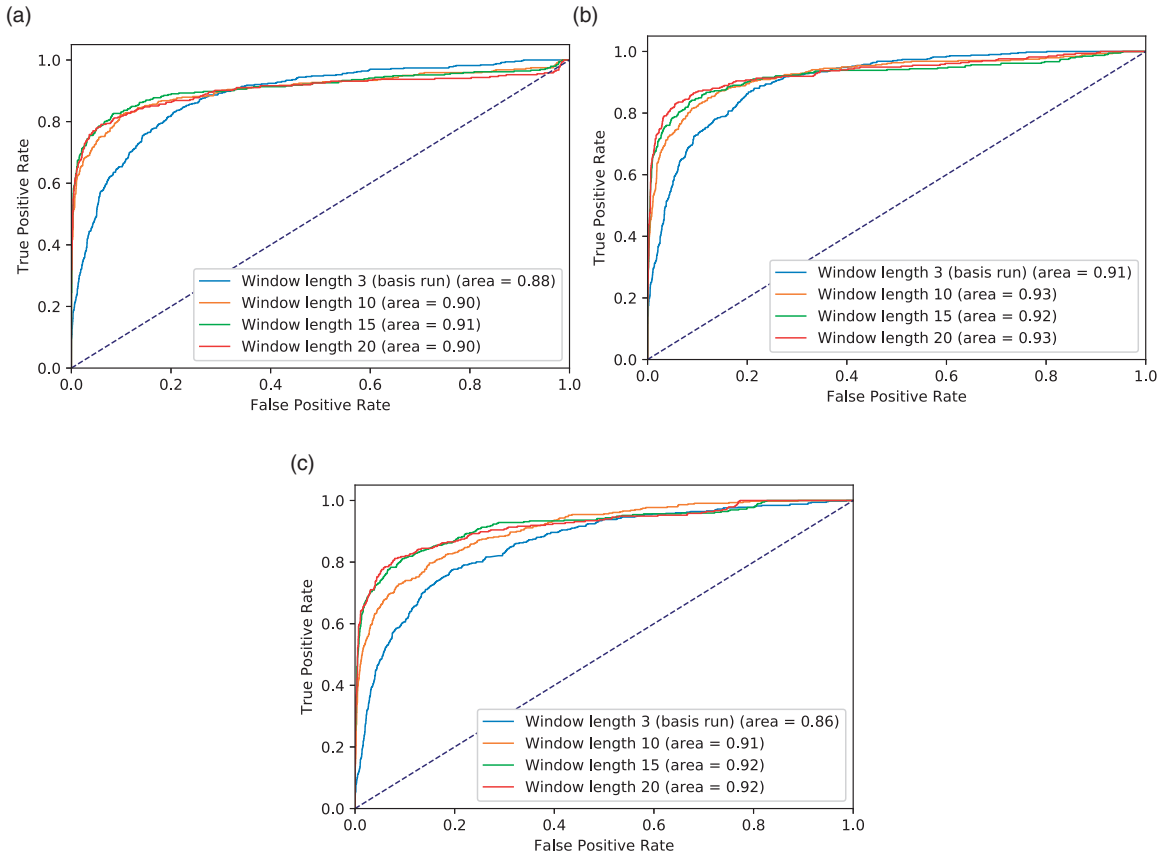

Figure 8. ROC curves showing the influence of the window lengths for each feature. In general, the increase in window length increases the performance, but it converges to the performance of a window length of 20 s. (a) MFCC-CNN for different window lengths. (b) Mel spectrogram-CNN for different window lengths. (c) Spectrogram-CNN for different window lengths.

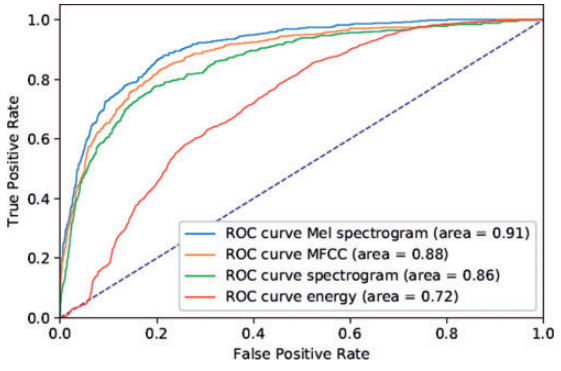

Figure 9. ROC curves of each feature for the basis run. Also the energy of the signal is used as an input for the ROC curve to show that the model does not base its output only on the energy in the signal. The Mel spectrogram is the best performing feature, MFCC second best, the spectrogram is the worst feature and energy performs significantly worse than all features.

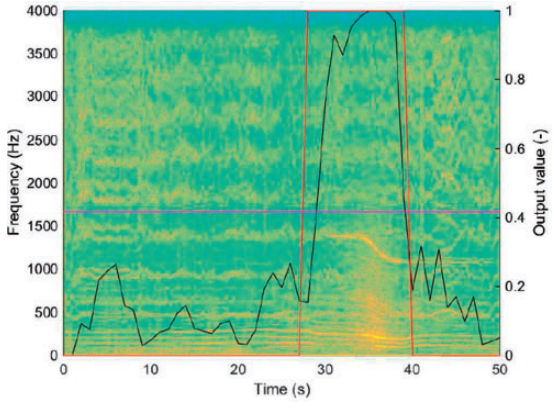

Figure 10. Correct classification example of a sound sample. In red is the expected label, in black the given output and in purple the discrimination threshold. The left axis belongs to the spectrogram only, the right axis belongs to the output, the label and the threshold lines. As the output is always under the purple line when the label is 0 and above the purple line when the label is 1 (except for 1 s), this sample is accurately classified.

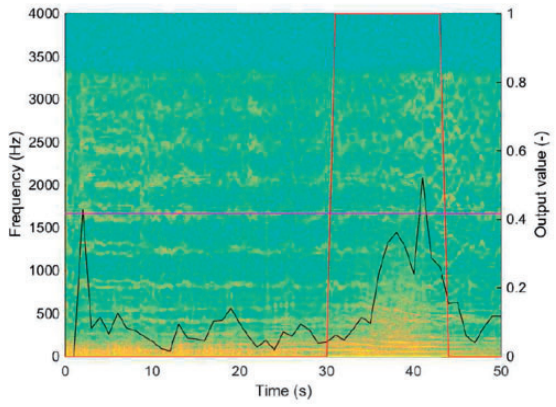

Figure 11. Partly wrong classification example of a sound sample. In red is the expected label, in black the given output and in purple the discrimination threshold. The left axis belongs to the spectrogram only, the right axis belongs to the output, the label and the threshold lines. A false positive is shown at 3 s and false negatives between 30 and 45 s (except second 40).

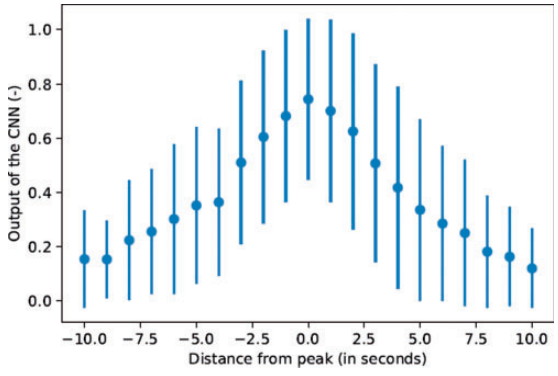

Figure 12. Means (dots) and standard deviations (bars) per time distance from the center of a CPA in the spectrogram. It shows that the closer the aircraft is, the better the detection performance.

Table 4. Precision and recall of the Mel spectrogram-CNN using window length 20.

	Precision	Recall
0	0.97	0.99
1	0.85	0.70

Table 5. Confusion matrix of the Mel spectrogram-CNN using window length 20.

		Predicted class	
Actual class		0	1
	0	2823	42
	1	101	234

ROC curve just separates from the Y-axis is chosen (which is around an FPR of 0.01 and a TPR of 0.7).

Discussion

The results shown in Section Results are further discussed in this section. Starting with the different UAV/aircraft amplitude ratios, Figure 5 shows in the lower FPR region an expected trend, which is that the lower the UAV amplitude is compared to the aircraft amplitude, the better the aircraft is detected. That means, in order to use this model for real-world application, it is best to diminish the UAV's ego-sound as much as possible, for example by means of the method of Marmaroli et al.[8]

The addition of third party database recordings also improves the performance, such as shown in Figure 6. Those recordings consist of different background noise, which could be easier for the model to distinguish from the typical background noise from the Lelystad recordings. The basis run performed better in the very low FPR (<0.01), but the corresponding TPR is too low to be a good detector.

The fact that the different type of labeling performs worse, which is shown in Figure 7, is unexpected. The labels that are one for the distant detection labeling consist of the ones from nearby detection labeling plus some extra ones before and after. In other words, the nearby detection labels are a part of the distant detection labels. As the distant detection labeling includes the nearby detection labels, it is expected that training with distant detection labeling at least performs the same as training with nearby detection labeling. However, the model performs worse (or equal, for any FPR lower than 0.05) which means that there is no benefit in using the distant detection labeling. The consequence of using nearby detection labeling over distant detection labeling is that the aircraft is closer to the UAV when it is detected.

The trends shown in Figure 8, at which the window length is increased, are not unexpected. The longer the window length, the more information the model uses to make a decision and therefore the performance is better. This only works up to a certain amount since sound information too far in the past can have nothing to do with the present sound. Based on the presented experiments, a window length between 15 and 20 s should be used to be as accurate as possible. Choosing a value above 20 s will not increase the performance and makes it computationally more expensive. Of course, also other forms of memory can be explored, such as long short term memory [21] or GRU.[22]

In the ideal situation, no false positives or false negatives are present in the output of the detector.

Since the ROC curves in Figures 5 to 8 never have an AUC of 1, this is not possible. Therefore, we aim to have as little false positives and false negatives. In Tables 4 and 5, a limit of one false positive in 100 s is set. If after a false positive a warning is send to the operator, once in 100 s he/she has to check whether there is really other air traffic present, which is not increasing the workload too much and therefore once in a 100 s is a reasonable limit. If the UAV has to descend (or even land) after a detection, a false positive once in a 100 s is already a lot. So for those cases, a filter should be applied which checks whether multiple positive detections are found in a short-time frame. The percentage of missed detections corresponding to this false positive rate is 30%. Luckily, Figure 12 shows that the closer the aircraft is, the better the accuracy, so the missed detections will mostly appear in the early stages of the detection.

Alongside the conclusions drawn from the results, there are a few general comments to be made concerning the research method.

Firstly, the data set should be extended. The recordings at Lelystad[12] were performed on a single day and single location. This can limit the types of background noise contained in the data and can be of influence to the results. The data set used in the basis case (run 3) only contains the recordings from Lelystad airport. This data set has in total 84 flyovers. The data augmentation increases the data set times four, so 336 flyovers are available for the data set. This is considered a relatively small data set for machine learning purposes such as this research. For comparison, ImageNet,[g] a famous data set for image recognition, has 15 million examples in total. In addition, the ratio of the data set that includes aircraft sound and that only includes background noise is not 50/50, due to the fact that the cut-outs from the recordings are random. The ratio aircraft/background in this data set is approximately 20/80. The problem with this ratio is that the model could classify all the sound samples as background noise and still would have an accuracy of 80%. Another comment about the data set is that it is artificially mixed, so the UAV and aircraft sound are individually recorded. In the spectrogram, it is visible where the aircraft sound is added to the UAV sound by vertical lines at the stop and start. An example is shown in Figure 13, at which the aircraft recording part stops at 30 s. In order to avoid this effect, recordings should be taken on a UAV, which flies close to flying aircraft.

So far, the only different scale used is the Mel scale. Two features use this scale which mimics the way humans perceive frequency. The comparison of the Mel spectrogram and the spectrogram in Figure 9 shows that stretching the lower frequencies works well in combination with the CNN. One idea is to make a scale that stretches the lower frequencies even more. As most of the distant aircraft sound lies in the low frequency region, further stretching the lower frequencies could show more important low frequency sound information for the CNN.

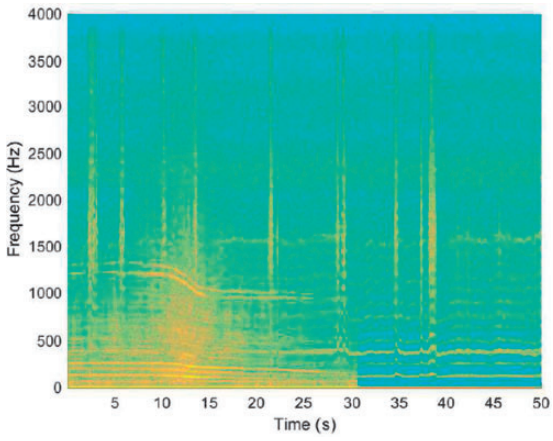

Figure 13. Spectrogram of a mix of UAV and aircraft sound. The end of the aircraft sound recording is visible on the spectrogram at 30 s by the vertical line (which is the sudden decrease in energy).

What is more, is that there is not much difference in type of background noise. Only two types of microphones are used, the 808 micro camera microphone and the microphone from the array. Different microphones could show different noise content. Further research in the quality of the microphones is demanded. Also, the background noise is pretty constant during the recordings, whereas on a flying UAV this could differ considerably. Other background noise, such as cars, trains, lawnmowers, etc., is not added.

Not only is there one composition of background noise, but also only one type of UAV sound has been used. In order to make a model for versatile applications, multiple UAV sounds should be included in the data set. If the model is applied to only one UAV, it is useful to use its specific model in training the detection network. In this process it is also important to check whether the ego-noise of the UAV is in the same order of loudness as the Parrot Bebop used in this research.

The fixed microphone array on the runway with overflying aircraft, also limits the types of geometries in the dataset. In a practical scenario, the UAV could be at different position with respect to the aircraft.

Last but not least, the types of aircraft to detect are limited. Differentiation between jets, propeller aircraft, helicopters or ground vehicles can be of great importance to the UAV in order to decide upon the best course of action.

Conclusion

Detection of air traffic sounds on a UAV could increase the safety of the airspace. This paper builds on existing sound features and classification methods, but this time applied to combine UAV and aircraft sound.

The three features used are the MFCC, spectrogram and Mel spectrogram, which are the input to a CNN classifier. The best performance of the model is obtained using the Mel spectrogram, which moves over the sound recording with a 20-s window length. The detection performance increases when the aircraft is closer to the UAV. Longer time windows give better performance up until a certain window length, but also decrease the potential reaction time for an avoidance maneuver. Secondly, the model works best if as little UAV sound is present as possible. Thirdly, the current method still gives too many false positives for real-world application. Improvements may be expected from a better filtering over time (ignoring solitary peaks of the network's output), a more extensive data set, and potentially additional information such as the commanded RPMs of the UAV's propeller(s). Finally, a more realistic data set should include sound recordings of aircraft taken from a (moving) UAV.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support forthe research, authorship, and/or publication of this article: This project has received funding from the SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 763702.

ORCID iD

Christophe De Wagter https://orcid.org/0000-0002-6795-8454

Notes

a. www.percevite.org
b. http://www.chucklohr.com/808/
c. https://freesound.org/
d. https://www.youtube.com/watch?v = uprXhH6-FNI
e. http://airportnoiselaw.org/dblevels.html
f. http://research.cs.tamu.edu/prism/lectures/sp/l9.pdf
g. http://www.image-net.org/

References

1. Tijs E, de Croon G, Wind J, et al. Hear-and-avoid for micro air vehicles. In: *Proceedings of the international micro air vehicle conference and competitions (IMAV)*, Volume 69, Braunschweig, Germany, 6-9 July 2010.
2. de Bree HE and de Croon G. Acoustic vector sensors on small unmanned air vehicles. Presented at *the SMi unmanned aircraft systems,* UK, November 2011.
3. Basiri M and Schill F. Audio-based relative positioning system for multiple micro air vehicle systems. In: *Robotics: Science and Systems RSS2013*, Berlin, Germany, 24-28 June 2013, pp. 1–8, http://www.roboticsproceedings.org/rss09/p02.html
4. Basiri M, Schill F, Floreano D, et al. Audio-based localization for swarms of micro air vehicles. In: *2014 IEEE international conference on robotics and automation (ICRA)*, Hong Kong, China, 31 May-7 June 2014, pp. 4729–4734. Piscataway, NJ: IEEE.
5. Basiri M, Schill F, Lima P, et al. On-board relative bearing estimation for teams of drones using sound. *IEEE Robot Autom Lett* 2016; 1: 820–827.
6. Basiri M, Schill F, Lima PU, et al. Robust acoustic source localization of emergency signals from micro air vehicles. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, 7-12 October 2012. IEEE.
7. Harvey B and O'Young S. Acoustic detection of a fixed-wing UAV. *Drones 2018;* 2: 4.
8. Marmaroli P, Falourd X and Lissek H. A UAV motor denoising technique to improve localization of surrounding noisy aircrafts: proof of concept for anti-collision systems. In: *Acoustics 2012*, https://hal.archives-ouvertes.fr/hal-00811003/ (accessed 29 January 2021).
9. McLoughlin I, Zhang H, Xie Z, et al. Continuous robust sound event classification using time-frequency features and deep learning. *PLoS One* 2017; 12: e0182309.
10. McFee B, Raffel C, Liang D, et al. librosa: Audio and music signal analysis in python. In: N Morgan, P Georgiou, S Narayanan, F Metze (eds) *Proceedings of the14th python in science conference*, 8-12 Sep 2016, pp. 18–25. ISCA.
11. Zhang H, McLoughlin I and Song Y. Robust sound event recognition using convolutional neural networks. In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, South Brisbane, QLD, Australia, 19-24 April 2015, pp. 559–563. Piscataway, NJ: IEEE.
12. Wijnker D, van Dijk T, Snellen M, et al. Hear-and-avoid: acoustic detection of general aviation aircraft for UAV, https://hdl.handle.net/10411/VXZ1AQ (2020, accessed 28 January 2021).
13. Doljé S. *Quantifying microphone array directivity*. Master's Thesis, Delft University of Technology, Delft, 2017.
14. McFee B, Raffel C, Liang D, et al. librosa: Audio and music signal analysis in python. In: Proceedings of the

14th *python in science conference*, Austin, Texas, 6–12 July 2015, pp. 18–25. SciPy.org.
15. Logan B. Mel frequency cepstral coefficients for music modeling. *ISMIR* 2000; 270: 1–11.
16. Lyon RF. Machine hearing: an emerging field. *IEEE Signal Process Mag* 2010; 27: 131–136.
17. Zheng F, Zhang G and Song Z. Comparison of different implementations of MFCC. *J Comput Sci Technol 2001;* 16: 582–589.
18. Chollet F. Keras, https://keras.io (2015, accessed 29 January 2021).
19. Abadi M, Barham P, Chen J, et al. Tensorflow: a system for large-scale machine learning. *OSDI* 2016; 16: 265–283.
20. Kingma DP and Ba J. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, https://arxiv.org/abs/1412.6980 (2014, accessed 29 January 2021).
21. Sainath TN, Vinyals O, Senior A, et al. Convolutional, long short-term memory, fully connected deep neural networks. In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, South Brisbane, QLD, Australia, 19-24 April 2015, pp. 4580–4584. Piscataway, NJ: IEEE
22. Cakir E, Parascandolo G, Heittola T, et al. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transac Audio Speech Lang Process* 2017; 25: 1291–1303.