Investigating signal processing techniques for multi-class sound identification

Kajol Vaitha

Department of Engineering Mathematics
University of Bristol

Project Report submitted in support of the degree of Master of Engineering

Supervised by Dr Martin Homer

April 2024

Abstract

This investigation explores various signal and image processing techniques for sound classification. We are motivated by species identification, from bird songs and calls, where one of the most common issues is extracting useful information from limited datasets and complex field recordings. The effects of the choice of techniques are investigated for spectrograms and mel-frequency cepstral coefficients (MFCCs) as inputs to convolutional neural networks. Each classification pipeline is expanded with morphological filtering operations and spectral subtraction, which aim to tackle the challenges of noise reduction and feature enhancement. Optimisation of these pipelines is further explored by examining the effects of varying the method parameters such as the type of structuring element used in morphological filtering, the number of MFCCs extracted, and the strength of noise reduction for spectral subtraction. The techniques are applied in combination with each other to gain an understanding of the processing requirements for classification tasks using noisy source data. After applying and evaluating the methods on three datasets with varying numbers of classes and levels of signal-to-noise ratio, it is found that MFCC extraction paired with the morphological operation of closing and spectral subtraction yields the most significant improvements compared to using the raw input. The study additionally offers considerations for further work, including the exploration of alternative signal transforms to the Fourier transform, using different machine learning models and image processing techniques. Overall, this study highlights the importance of noise reduction in sound classification tasks and the potential that image and signal processing techniques have for tackling the common challenges in this field, particularly for bird sound identification.

Ethics statement

This project does not require ethical review as determined by my supervisor Dr Martin Homer. No new data was gathered from any human or animal participants. The datasets used in the project are all publicly available, contain no human identifiable or sensitive personal data, and were used within the terms of their license, as noted in the text.

Contents

1 Introduction

Sound classification is a task that spans multiple applications such as environmental monitoring, health care and speech recognition. The task of distinguishing different types of sound has been an ongoing area of interest in signal processing and machine learning. There have been many developments and applications that can perform sound identification for music and animal species. Real-time sound recognition is becoming more popular, with applications such as Shazam and Merlin ID showing huge potential for identifying songs or bird species from their environment. Through a combination of spectrogram image processing techniques and convolutional neural networks (CNNs), Merlin ID provides real-time suggestions of the birds that can be heard from the user's recording of their surroundings [1].

There are many challenges faced by these algorithms in achieving good quality sound classification. One of the main obstacles is the need for a large and representative database to learn from. Learning improves with each new recording, however, it is difficult to obtain a large enough number of recordings to guarantee a high level of accuracy for each class. It can be difficult to identify the particular segments of the recordings which contain the most useful information, especially if there are multiple identifiable sounds in one clip. The level of difficulty depends on the type of sounds. For songs, there are standardised musical structures which aid the classification through known rhythms, chords and melodies. However for bird species, the vocalisations will inherently have some variation, for example across individuals in the same species, and requires a more adaptive approach to classification. This requires good labelling of the data which is typically carried out by ornithology experts, who can identify the species from diverse environments.

These issues stimulate the investigation into improving the existing signal processing pipeline to identify species with limited information. The aim of this project is to investigate different approaches to improve the existing methods of signal processing. With a broad focus on bird sound identification, the effects of the methods on the classification performance metrics will be explored. The primary focus will be placed on how to extract useful information from the audio signals before they are input to a machine learning model. The use of signal, sound and image processing techniques such as mel-frequency cepstral coefficients for feature extraction and morphological filtering for image processing are discussed and evaluated, in order to gain insights into areas of interest and improvement for the sound classification process motivated by bird species identification.

In Chapter 2, current methods for sound identification are explored, establishing the techniques that will be applied. Chapter 3 introduces the classification pipelines and datasets used in the investigation. In Chapters 4 and 5, data processing techniques are added to the pipeline to investigate their effects on classification performance and the processing requirements that must be considered. Parameter sensitivity is explored in Chapter 6, where the effects of the parameters in the processing step are investigated. Chapters 7 and 8 discuss the performance and insights gained from the investigation, proposing avenues for further extensions.

2 Background

2.1 Current methods for bird sound identification

There are some existing bird identification apps such as Warblr, iBird and eBird which are used by birdwatchers to track and identify sightings of bird species. The Merlin ID application is one of the few established and accessible apps that has the ability to identify birds from their sounds. It can identify 458 bird species in the US and Canada, and around 375 species within Europe [2]. The classification pipeline involves taking the recording of the sound from a mobile phone, converting the audio file into a spectrogram, and inputting this image to a CNN, trained by gradient descent.

The Merlin neural network is trained on 140 hours of bird audio and 126 hours of non-bird background audio, such as whistling and car sounds. This data has been labelled by sound identification experts from the eBird community and Macaulay Library to identify the exact moments the bird sounds appeared in each clip, matching them to the correct species. The model is trained to identify species from audio recordings it has not encountered before, with each new recording contributing to the bi-annual model update. Merlin ID is continually being improved to support the species it can currently identify. It expands its capabilities to identify new species, however there must be enough recordings to train with before doing so. Although Merlin takes inspiration from other bird sound classification projects such as BirdVox and BirdNET, the authors claim that their application performs better [3, 4]. This is due to the labour-intensive process of labelling the precise moments in each sound recording where the particular species are heard, to avoid erroneously classifying multiple sounds as one species. This creates the challenge of investigating methods that could automate or replicate this process using alternative sound processing and feature extraction techniques, without the need for manual labelling by experts.

The input to the CNN used by Merlin is a spectrogram, a visual representation of the sound in terms of frequency and time. Spectrograms have a colour scale corresponding to the magnitude of the time-dependent frequency components. They are commonly used in sound classification due to the large amount of information they contain. The first stage for generating a spectrogram is to split the audio signal into equally sized segments in time and compute the Fourier transform (FT) of each segment. Usually these segments overlap due to windowing, a process which mitigates the effects of spectral leakage. Spectral leakage is an effect caused by the mismatch of amplitudes at the ends of a non-integer period signal [5]. There are multiple types of windowing functions that can be applied, of which the Hanning function is the most common [6]. The FTs for all of the segments are combined into a plot of frequency against time.

There are several considerations to make when creating the spectrogram, each with their own trade-offs relating to time and frequency resolution. One of the key decisions here is to re-scale the frequency to the mel scale. Raw spectrograms, without scaling, are often uninformative in terms of the data they display. This is due to the way humans differentiate pitch and amplitude. The perception of pitch for humans is not linear; we are more sensitive to differences in lower

frequencies than higher frequencies. This concept was introduced by Stevens and Volkmann in 1937 [7]. The conversion between the frequency (f) in Hz to the pitch (m) in mels is given by $m(f) = 2595 \log\left(1 + \frac{f}{700}\right)$. Similarly, humans perceive amplitude logarithmically, accounted for by the decibel scale. The mel spectrogram uses these two key differences, plotting the frequency in the mel scale and amplitude using the decibel scale, to better represent the perception of the audio. Extracting features relating to the mel scale is more commonly used for sound classification tasks and have been found to have improved performance compared with the use of the Hz scale [8].

2.2 Music identification

There are many other applications that perform similar sound classification tasks. One of the better known is Shazam. Created in 2002, the application is now one of the most widely used song recognition platforms [9]. Similar to Merlin ID, Shazam uses the Fourier transform in windowed segments. The peaks of each segment are identified as the frequencies with the highest magnitude, forming the signature for that section of audio. Each song has a unique hash tag, containing information about the time that the set of identified frequencies appear. A database of songs is searched for matching hash tags when a song is being identified in the app [10]. However, many songs will have some matching hash tags due to similarities in chord progressions and sampling of other music used within a different song. To overcome this, Shazam analyses and matches the relative timestamps of the frequencies and the sample to narrow down the possible songs. The final set of possibilities are ordered by likelihood to suggest the matching song to the user.

Google's Hum to Search (GHTS) is another sound recognition tool that identifies possible songs based on a 10-15 second clip of the user humming a tune. In a similar fashion to Shazam, each song melody has a unique and identifiable fingerprint. GHTS transforms the given input audio to a number-based fingerprint, based on its spectrogram representation and compares it to the database of learned songs [11]. In 2017, Google launched Now Playing, which builds upon the methods used in GHTS [12]. The musical features of the eight-second clips of audio are projected into low-dimensional embedding spaces of 7×2-second clips with overlapping one-second intervals. The song database, generated by a neural network, is searched to find similar embedded sequences. The first step of this is performing a nearest neighbour search on the database to search through millions of embedding vectors [13]. The audio is likely to contain noise, as well as inaccuracies in key, tempo and rhythm, so this initial search is approximate. To refine the possible songs, a similarity score is calculated between the sample and each option, strengthened by using a sequence of embeddings to reduce the amount of false positives. The hum to search model faces the challenge that the hummed version of the songs could be vastly different to the original, therefore the database of inputs from which the model learns can be limited. Similarly to Merlin ID, the limited amount of data restricts the ability to accurately detect the correct song and the database requires manual updates.

2.3 Wider sound classification

Sound classification tasks extend beyond the scope of bird sounds. In the medical field, the use of signal processing and classification has aided the diagnosis of heart-related diseases from stethoscope signals. Zeinali et al. investigated several signal processing algorithms to classify three types of heart sound that are used to diagnose cardiovascular disease [14]. The wavelet transform is applied to the signal, decomposing the signal into wavelets. The wavelets are oscillations localised in space and time. This transform is an improvement to the Fourier transform as it provides information in both the frequency and time domain.

Other applications within the medical field include classification of lung sounds to further evaluate the diagnosis of pulmonary diseases. Bardou et al. compare the use of spectrograms, MFCC statistics and local binary patterns extracted from spectrograms as inputs to the CNN [15]. The MFCC statistics method extracts six features including the mean standard deviation of the extracted coefficients which are passed to the classifier. The binary pattern method converts the spectrograms into binary images for classification. The images have a high dimensionality, reduced by principal component analysis. It was found that the CNN method performed the best compared to feature-based approaches such as support vector machines and k-nearest neighbour classifiers.

2.4 Automatic speech recognition and MFCCs

Automatic speech recognition (ASR) is a widely researched area, which is important for applications such as speech-to-text and hands free communication to improve accessibility. There have been many advancements in this field, particularly for real-time speech recognition from different speakers, languages and backgrounds [16].

A signal processing technique commonly used for speech recognition applications is the extraction of Mel-Frequency Cepstral Coefficients (MFCCs). In a comparative study of ASR techniques, it was found that MFCCs are the most widely used feature extraction method [17]. The MFCCs are computed by splitting the signal into overlapping short time frames, where 25 ms is standard [18]. The fast Fourier transform of each window is taken on each frame to obtain the power spectrum of the signal. Then, a mel-spaced triangular filterbank is applied to characterise the frequencies on the more representative mel scale. The filterbank is typically a set of 40 vectors, where the value within each vector is zero apart from a single, non-zero peak, illustrated in Figure 1. The filterbank energy is calculated by multiplying each of the vectors with the power spectrum, resulting in 40 values that represent the signal. The number of coefficients that are extracted can be chosen, where the maximum is usually 40. The final step is to take the log of the spectrum and apply the discrete cosine transform to decorrelate the coefficients, resulting in the MFCC representation of the signal.

Despite the use of MFCCs becoming the standard practice for ASR, there are some limitations to this method of feature extraction. The most important consideration is that MFCCs are not

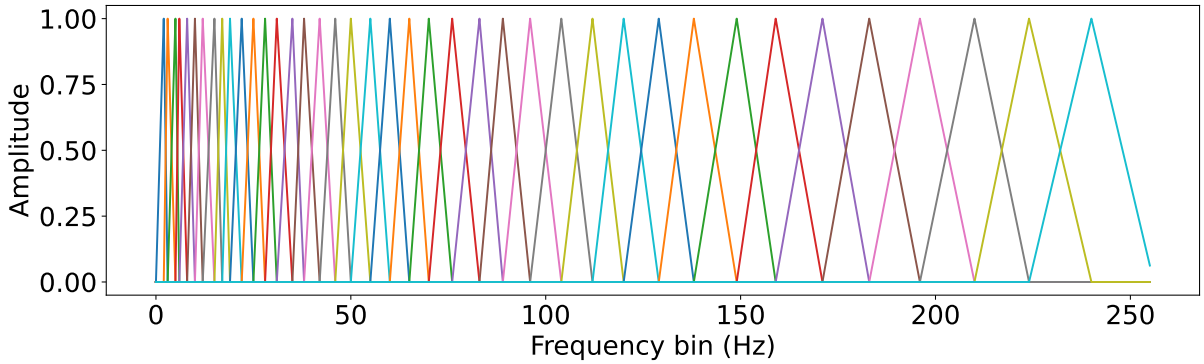


Figure 1: Example of triangular filterbank on the mel scale with 40 values indicated by each colour.

robust to noise, and distortion of the frequency will alter the entire feature vector [19]. From its calculation, the MFCCs are derived only from the power spectrum of the signal without considering phase. To mitigate this issue, past research applied feature enhancement before extracting the coefficients. For example, Korba et al. use the wavelet transform before obtaining the MFCCs, as well as adapting the MFCCs to also consider the phase spectrum [20]. The use of the wavelet transform during feature extraction has also performed well in various other sound classification tasks, for example types of animal and environmental sounds [21].

As identified through the literature, there are multiple challenges faced by the methods used in sound identification, such as ensuring important information is retained during processing. These could be overcome through the investigation and implementation of alternative signal processing techniques. We will use these ideas to conduct an investigative study on how different methods could be used to improve the performance of sound classification tasks, broadly linked to the application of bird species identification.

3 Demonstrative classification pipelines

3.1 Introduction

To gain a deeper understanding of the current challenges faced in sound classification, we describe in this chapter the creation and analysis of proof-of-concept pipelines. This allows for a baseline from which we can identify the potential points in the pipeline that can be improved or investigated further. In order to assess the capabilities of each method, we use three datasets which each demonstrate a particular concept under investigation. Each method will be implemented using these demonstrative pipelines and their results will be compared to the baselines, providing a means of evaluating the effects each technique has on sound classification.

The Free Spoken Digit Dataset (FSDD) contains 3000 recordings of spoken digits from 6 speakers [22]. Each digit (0 to 9) has 300 instances, 50 from each speaker. The sound files are trimmed such that there is minimal silence at the start and end of the recordings. This means that the part of the sound we wish to identify is completely isolated. The purpose of this dataset is to provide a way to evaluate the capabilities of the techniques on an ideal large, clean dataset and gather insights about how each method could be used to enhance the classification pipeline for noisier data.

The British Birdsong Dataset (BBD) contains bird sound recordings from the Xeno Canto collection [23]. The BBD contains 264 sound files, each labelled with their genus, species, location and type of sound such as songs, flight calls and alarm calls. There are 88 species of birds, with 3 sounds per species contained in the dataset. This is an example of a common problem in sound classification, where there are many classes to identify, but limited data to train models on [24]. The metrics obtained from this dataset will illustrate how each technique performs with a lack of instances during training, and we can draw insights into potential methods to deal with this issue. The BBD represents the most realistic case of having both limited and noisy data.

The ESC-50 dataset contains 2000 labelled sound recordings for 50 classes of sounds, with 40 clips per class [25]. The recordings have been taken from a wide range of sound events from the Freesound Project [26]. This includes sounds such as water pouring, glass breaking and crackling fires. This dataset provides an example of balanced data, with equal numbers of sound clips in each class. The motivation for using this dataset is that we can investigate how noisy input data is handled during classification. Within the task of identifying bird sounds, field recordings typically contain high levels of noise, including both other species and background sounds. Therefore it is important to consider how to effectively extract the relevant information from the raw audio clips. There are more sounds per class than with the BBD, making the ESC-50 dataset representative of a more ideal case, where there are more instances to use in training. We use this dataset as an intermediate case, where there are many classes of sound but with a cleaner set of raw audio.

We create two proof-of-concept pipelines to investigate the feature extraction and preprocessing

techniques. Figure 2 illustrates the top-level functions of the process. Initially, we create two pipelines for spectrograms and MFCCs as input to the machine learning model. Both pipelines will follow the same overall structure, with steps 1, 4 and 5 being consistent throughout. Steps 2 and 3, highlighted in red, will see changes as we make modifications to the methods used in generating the model inputs. In the spectrogram pipeline, we require additional preprocessing steps such as cutting the audio to specific lengths. However for the MFCCs pipeline, this is omitted. Step 3 in the process either involves generating spectrogram images or extracting MFCC feature vectors to represent the audio.

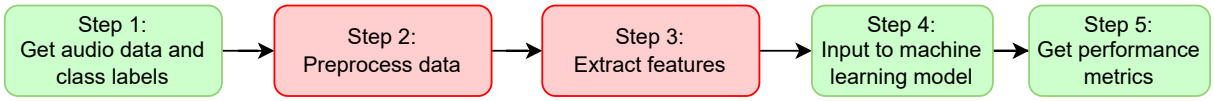


Figure 2: Steps in the demonstrative pipelines. Steps 1, 4 and 5 are unchanged with each pipeline iteration. Steps 2 and 3 are modified throughout the investigation.

3.2 Spectrogram pipeline

We first create a pipeline to demonstrate the process of generating spectrograms from audio signals to be input into a machine learning image classification model. As mentioned in Chapter 2, there are different types of visual representation of the sound, with spectrograms and mel spectrograms being the most common.

As an example, we take a bird sound of a common redshank birdsong from the BBD dataset, shown in Figure 3.

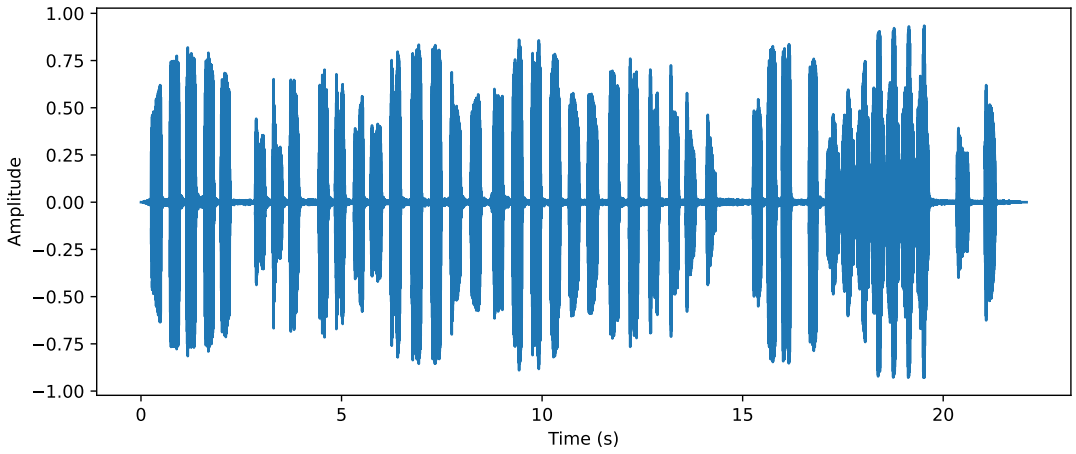
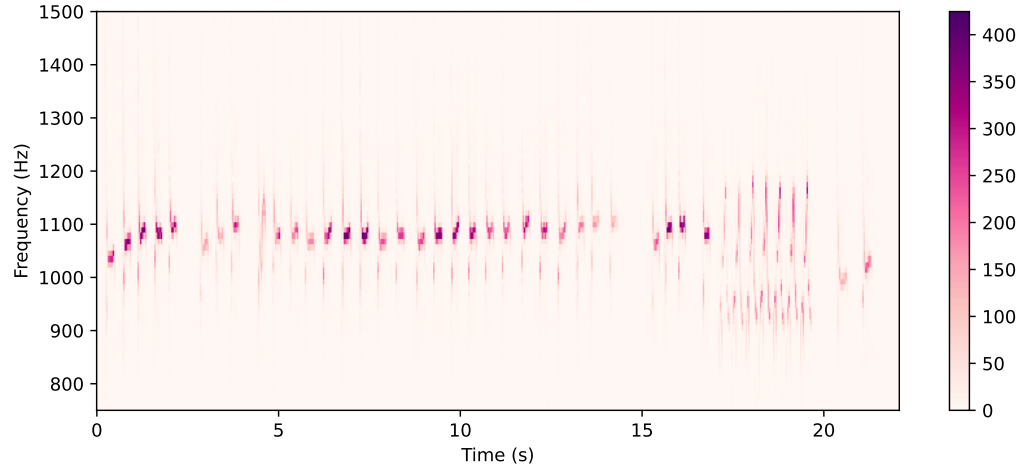


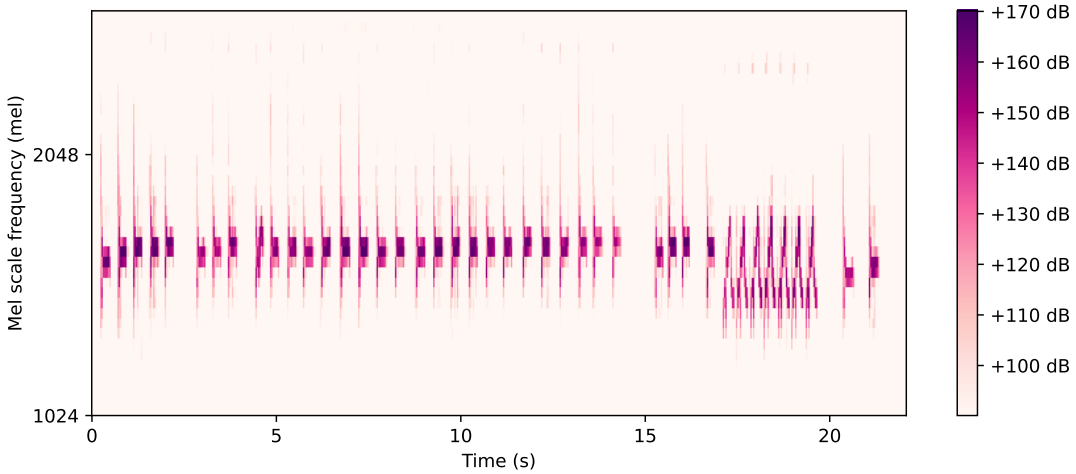
Figure 3: Time-domain visualisation waveform of sample bird song recording from BBD dataset.

We then generate the linear spectrogram and mel spectrogram of the sound. These are obtained using Librosa, a widely used Python package for music and audio analysis [27]. The audio data and sampling rate are extracted, where the sampling rate here is 44.1 kHz. The short-time Fourier transform (STFT) is applied, returning the representation of the signal in the time-frequency domain. A standard spectrogram is the visual representation of the STFT, shown in Figure 4a. To generate the mel spectrogram of the audio, the frequency in Hz is converted

to the mel scale and the amplitude is converted to the decibel scale, creating the spectrogram shown in Figure 4b. By applying these conversions, we see that the mel spectrogram shows a more informative representation of the sound clip, as the structure of the original spectrogram is retained while enhancing the significant areas of sound. This is particularly useful for the classification task as we require the key sound features to be as distinct as possible, making the recognition of patterns easier. If there is less distinction between the portions of the audio with more prominent amplitude compared to the background, there is a risk of erroneously classifying noise or missing out key characteristics of the spectrogram for identification. This demonstrates the benefits of using the mel and decibel scale in analysing and extracting information from the audio.



(a) Spectrogram of common redshank birdsong recording.



(b) Mel spectrogram of common redshank birdsong recording.

Figure 4: (a) Sample spectrogram from BBD with frequency in Hz. (b) Sample mel spectrogram from BBD with frequency and amplitude converted to the mel scale and decibel scale respectively.

The mel spectrograms are generated for each sound file in our datasets. Rather than the raw STFT output, the mel spectrogram images will be the inputs to the next step in the pipeline: classification using a machine learning model. For this task and throughout the investigation, we use a CNN as they are widely used for image processing and sound classification tasks [28].

CNNs require the images to have uniform size, so the spectrogram images must be resized before inputting them in the model. During exploratory analysis, we find the sound clips in the FSDD and BBD have varied lengths. There are a variety of ways to overcome the sizing issue, such as padding the smaller images out, or cutting the larger images down. However, this can cause problems in retaining the important data within the spectrograms. For demonstration, if we naively pad the spectrograms to be the same size as the largest file, we may change the structure of the data. For the BBD dataset, the largest file is significantly larger than the smallest file. We see that for padding out significantly shorter sound files, the important information is contained in only a small proportion of the resulting spectrogram, and the rest is zero or a constant value. This means there will be a bias towards larger images (i.e. longer sound recordings) during training.

Similarly, by cutting the spectrograms to the same size as the smallest image, we also risk losing information. We visualise this for our common redshank example. Figure 5 shows the resulting mel spectrogram cut to the smallest size of the entire dataset. Although the cut spectrogram alone may look informative in terms of the frequencies present in the call, the small portion of sound is taken out of context of the full birdsong, and we lose information about the structure of the audio or the time between each sound is made. Additionally, we notice that in the later portion of the original spectrogram between 17 s and 20 s in Figure 4, there is a change in the pattern. This is not accounted for in the cut version, and we lose this useful information entirely.

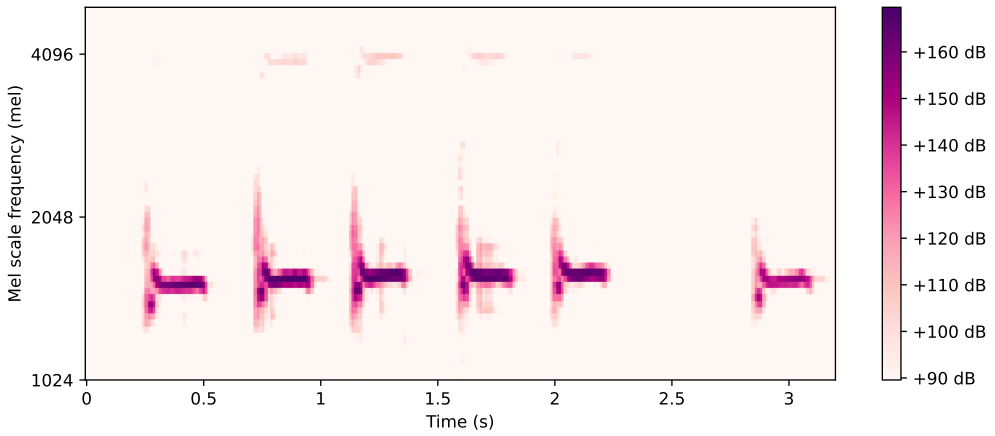


Figure 5: Sample mel spectrogram from BBD cut to the smallest size of the entire dataset.

Some sound recordings may have noise or silence at the beginning of the clip, before the main sound appears. It is therefore important to ensure that if cutting is used, a method is used to locate and extract the most informative parts of the audio. However, this can be a difficult task to perform manually, hence applications such as Merlin have experts that hand label and segment the audio in order to generate a larger dataset of identified calls for training [29].

Due to the varying sizes and characteristics of the generated spectrograms, we use a different CNN for the FSDD dataset as we do for ESC-50 and BBD. The recordings in the FSDD are significantly smaller in size and require a simpler CNN to be able to classify the sounds well. The ESC-50 and BBD recordings are longer, meaning the generated spectrograms are much

larger. This requires the CNN architecture to be different in order to handle the size of the inputs. This has been verified by initially training all datasets with the FSDD model, which yielded extremely low training accuracies for the ESC-50 and BBD datasets.

3.3 CNN design

For the FSDD, we use a fully connected CNN with four dense layers, illustrated in Figure 6. The input is a flattened, 1D vector of the original image. The model can handle the size of the generated vectors as the lengths of the audio clips in the dataset are relatively short.

The input layer uses the ReLU activation function, which is used to capture complex patterns from the input with 100 neurons. The second dense layer has double the number of neurons paired again with ReLU activation. This allows for the introduction of non-linearity and increases the capacity of the model to enhance the representation of the learned features. Before reaching the output layer, there is a final dense layer with 100 neurons to further refine the feature representation and capture more abstract attributes. The output layer contains the same number of neurons as classes, which for the FSDD is 10. The commonly used softmax activation function converts the raw predictions to a probability distribution for classification [30]. This neural network uses the categorical cross-entropy loss function and Adam optimiser, which has an adaptive learning rate that can lead to faster convergence and is regarded as the default algorithm [31].

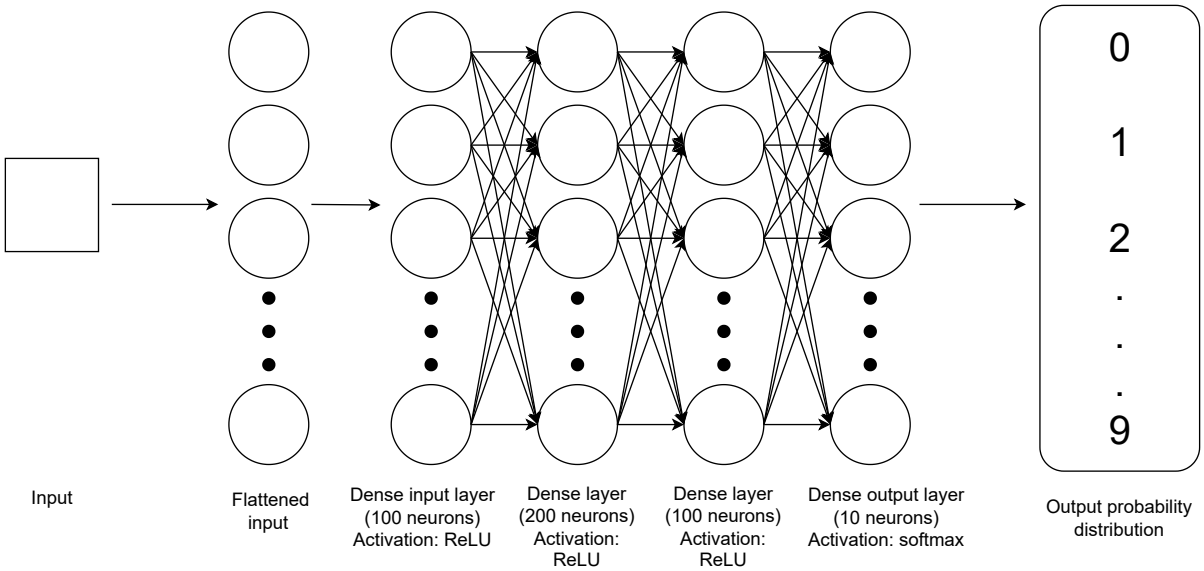


Figure 6: Fully connected CNN architecture for the FSDD dataset used to classify spoken digits from 0 to 9.

Figure 7 shows the CNN architecture used for classifying the sounds in the ESC-50 and BBD datasets. Due to the more complex nature and length of the audio recordings, the neural network required involves more steps than for the FSDD.

The key differences in this model are that the images are not flattened before entering the

input layer, and we use convolution and pooling to extract and learn the features from the data. Using a flattened input would be too large for the computing resources available, resulting in poor performance during training. For this CNN, the model takes inputs of size 128×282 and 128×431 for the ESC-50 and BBD datasets respectively. The input layer is a 1D convolutional layer with 32 filters, capturing local patterns and understanding the temporal nature of the sounds. Similarly, the second convolutional layer allows for more complex patterns to be identified, due to the larger number of filters. After each convolutional layer, we use a maximum pooling layer with size 2, which reduces the dimensionality and enables the model to be more computationally efficient and less prone to overfitting [32]. Before the dense layer, the feature map is flattened into a 1D vector. The fully connected dense layer with 128 neurons allows for further learning of the features, with the ReLU activation function providing non-linearity. The dense output layer, of size 50 for ESC-50 and 85 for BBD, uses softmax activation to output the probability distribution of the predictions. Again, we use the categorical cross-entropy loss function and Adam optimiser.

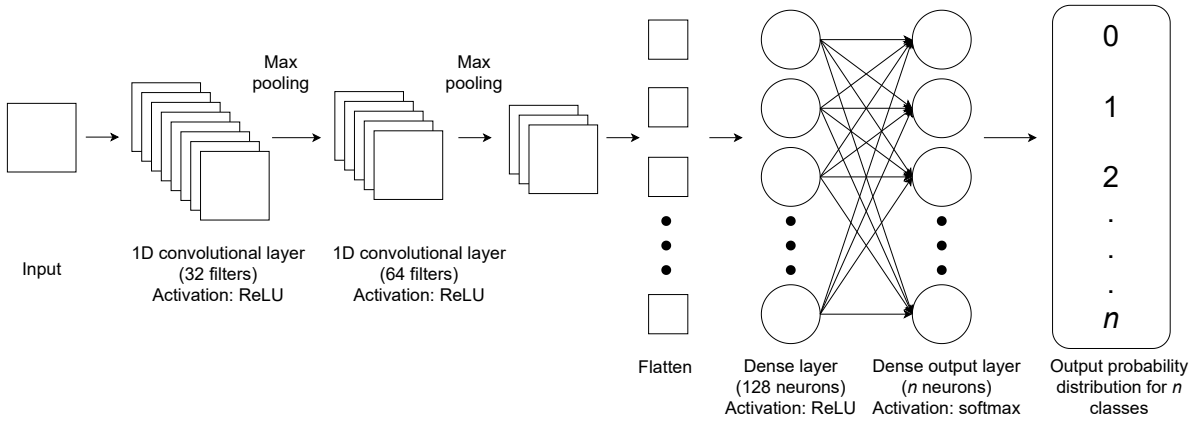


Figure 7: CNN architecture used for the ESC-50 and BBD datasets to classify $n = 50$ sound types for the ESC-50 dataset, or $n = 85$ bird species for the BBD dataset.

We split the data from each dataset into training, testing and validation with 80%, 10% and 10% of sounds in each group respectively. We apply the models to each of the three datasets to understand the performance on varying quality of data. Table 1 shows the results of the initial investigation for mel spectrograms. The FSDD, with relatively clean and noise-reduced sound clips performed the best, with an F1 score of 91.33%. The sound files in this dataset have varying length, however the difference between the length of the smallest and largest spectrogram is minimal. Cutting did not hinder the ability to capture the majority of information in the FSDD as much as it did for the BBD dataset, which has more widespread lengths of recordings. This is evidenced by the poor performance, achieving an F1 score of only 8.18%. Applying the mel spectrogram method to the raw ESC-50 dataset also performed poorly, despite the sound clips all being the same length. The F1 score of 28.51% highlights that there are other factors concerning the recordings that will affect the performance. One of the key characteristics of the ESC-50 dataset is that it contains recordings from noisy outdoor environments. Therefore the raw audio must be processed further in order to remove noise, and extract the most important features. Noise reduction and data manipulation techniques will be investigated in Chapter 4,

to identify areas for improvement within the classification pipeline.

Through this initial demonstration, we see how difficult it can be to process the audio to maximise the information we give to the machine learning model. Merlin ID uses the spectrogram image classification method to perform sound identification. This justifies the investigation into a more efficient and effective way to carry out audio processing.

	Training accuracy	**Test accuracy**	**F1 score**
FSDD	0.9513	0.9133	0.9133
ESC-50	0.9969	0.2825	0.2851
BBD	0.9905	0.0755	0.0818

Table 1: Performance metrics of the mel spectrogram classification pipeline, after cutting is applied to the FSDD and BBD datasets.

3.4 MFCCs pipeline

The MFCC signal processing pipeline uses an alternative method to spectrogram creation. Using MFCCs requires less consideration for the length of the sound files during preprocessing as the input to the CNN is a fixed length vector of features instead of an image. We use the Librosa function in Python to extract the MFCCs introduced in Chapter 2.4. The number of coefficients to return, n_{MFCC}, is the key parameter for the function, meaning that the output will be the same size for all files, regardless of the length of the sound clip.

The audio time series and native sampling rate are extracted for each clip. We perform the MFCC extraction on each dataset, with $n_{\text{MFCC}} = 40$, the maximum number of coefficients that can be extracted. This value has been chosen as a baseline and its effects on the quality of performance are investigated in Chapter 6.3. For this pipeline, we fix the value at 40 so we can have a more discriminative feature space. This could be beneficial for audio data in which there are subtle differences between classes. For example, the ESC-50 and BBD datasets require a feature representation that is robust to distortions and small variations.

The neural network takes the 40-dimensional vectors of features as input for model fitting. After the model is trained and tested on each of the three datasets, we obtain the results shown in Table 2. Figure 8 shows a comparison of the baseline pipelines. We see that the F1 score improved from the spectrogram pipeline for all datasets, indicating that at a base level, the MFCC method is a step towards bettering the classification pipeline. As shown in Table 2, the ESC-50 dataset has a training accuracy of 99.81% and test accuracy of 41.01%. For the BBD dataset, we obtain 99.72% for training accuracy and 20.82% for test accuracy, indicating overfitting during training. Since the model is overfitting the data, an investigation into the effect of the value of the n_{MFCC} parameter is a possible aspect to explore model performance.

The resulting performance is impacted by the quality of the dataset. We see that the FSDD, a dataset with 300 instances of each label, performs very well in comparison to the ESC-50 and BBD datasets, which only have 40 and 3 per class respectively. We require an investigation into

	Training accuracy	Test Accuracy	F1 score
FSDD	0.9948	0.9512	0.9512
ESC-50	0.9981	0.4101	0.4080
BBD	0.9972	0.2082	0.2092

Table 2: Performance metrics of the initial MFCC classification pipeline for the FSDD, ESC-50 and BBD datasets.

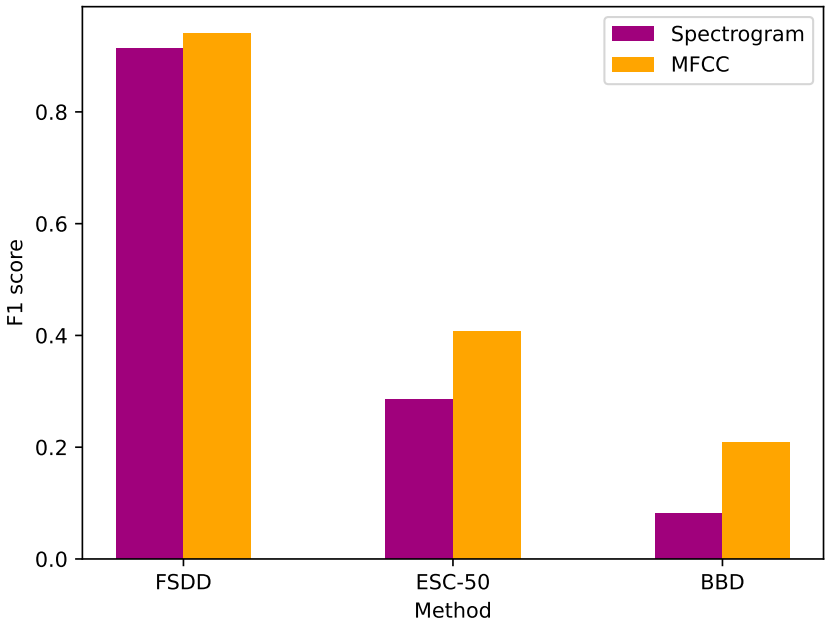


Figure 8: F1 scores of the mel spectrogram and MFCC classification pipelines to be used as baseline results.

maximising the amount of information we extract from the limited raw data we are given. The FSDD is also the least noisy dataset, which suggests that noise may also be a key factor in the performance of the model. The models incorporate the background sounds during training. The methods described so far do not distinguish the sound from the background noise. As discussed in Chapter 2.4, one of the limiting factors of MFCCs is their lack of robustness to noise [19]. This stimulates exploration into noise reduction in the signal processing step in Chapter 4.

4 Data processing techniques

The performance metrics of the FSDD compared to ESC-50 and BBD datasets highlight the importance of having clean sound files to input to the machine learning model, evidenced by the significant difference in F1 scores. This stimulates an investigation into the preprocessing stage of the pipeline, particularly in removing unwanted noise from the raw audio signal.

We evaluate methods identified in the literature to assess their potential for our task. We investigate morphological filtering techniques and spectral subtraction to filter and manipulate the audio signals and provide the neural network with a more informative input [33, 34].

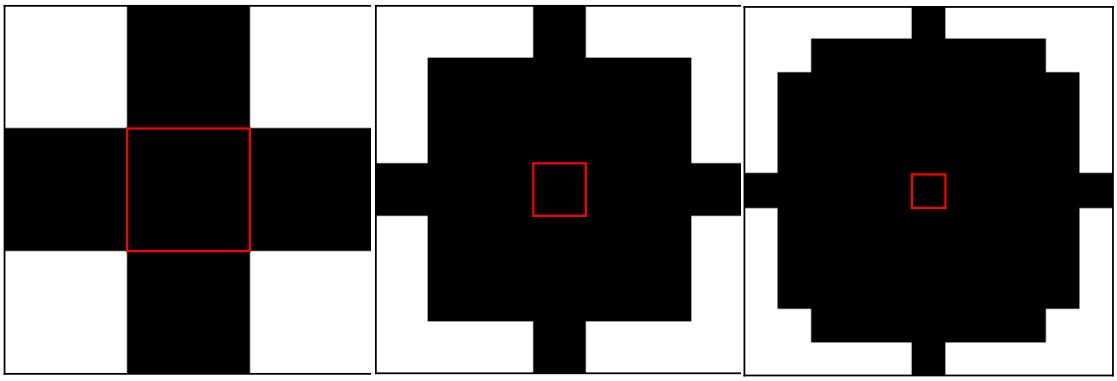
4.1 Morphological filtering

4.1.1 Methods

Morphological filtering (MF) is a set of nonlinear operations used for image processing. There are four standard morphological functions which we consider: erosion, dilation, opening and closing. This form of image processing is used to change the shapes of the objects within an image by shrinking, expanding, smoothing and correcting features [35]. MF is commonly used in feature extraction for automatic speech recognition (ASR) tasks. In particular, there have been studies on using techniques that replicate how the human auditory system is able to process and identify sound [36]. MF is commonly used in image processing, for enhancing structure, defining shape boundaries, separating the foreground from the background and filtering noise [37]. We apply some of the techniques to our three datasets to evaluate the potential benefits MF can have in audio classification.

Each of the methods involves passing a structuring element (SE), or mask, over the spectrogram. The choice of SE defines the resulting output, and can take any shape or size. The SE is applied to each pixel and its output is determined by the neighbouring pixels within the shape of the SE and the type of MF being applied. There are a variety of shapes that can be used, such as a disk, square or star. Each shape will alter the image in different ways, and should be chosen based on the underlying characteristics of the data. The size of the elements are defined by a single parameter of width or radius. Each pixel in the image has a value corresponding to the amplitude. A smaller sized element considers fewer pixels when determining their values in the new image. Figure 9 illustrates how the shape of the disk SE changes with the radius parameter. In the demonstrations of each MF function, we use a default SE of a disk with radius 1. This parameter value is chosen to preserve finer details in the images [38]. The size and shape of the SE is an aspect that can be further investigated in our study.

Erosion is used to shrink objects in an image by removing pixels at the boundaries. The pixel under consideration is changed to the minimum value of the pixels within the structuring element around it. The erosion operation is a convolution, defined by Equation 1 [36]. For two-dimensional signal $S(f,t)$ and structural element $M(f,t)$, the erosion, $\ominus$, of S using M is



(a) Disk with radius 1 (b) Disk with radius 3 (c) Disk with radius 5

Figure 9: Visualisation of the disk structuring element with radii of (a) 1, (b) 3 and (c) 5, with the centre pixel highlighted in red.

given by

$$(S \ominus M)(f,t) = \bigwedge_{(\varphi,\tau)\in\mathbb{R}^2} \{S(f,t) - M(f-\varphi, t-\tau)\}, \tag{1}$$

where $\bigwedge$ is the min operation. The resulting image is a reduced or shrunken version of the original, with small areas of noise removed. In Figure 10, we visualise an example from the ESC-50 data to illustrate the MF functions. We use the example of church bells ringing, where we see the distinct moments where the bells sound. We first take the time waveform of the sound and generate the mel spectrogram, shown in Figure 10a and Figure 10b respectively. The resulting image after erosion is seen in Figure 10c. We see that the shapes are less prominent and there are more gaps in the spectrogram compared to the original. Areas with lower amplitude are considered to be unwanted noise, and are removed from the spectrogram image. This is particularly noticeable in the upper half of the image, where the signal at the higher frequencies have been eroded.

Dilation is the dual function of erosion. This process expands the boundaries of objects in the image, filling small gaps between them. Each pixel is changed to the maximum value of the pixels that lie within the coordinates of the SE around it. The dilation operation, $\oplus$, is defined by

$$(S \oplus M)(f,t) = \bigvee_{(\varphi,\tau)\in\mathbb{R}^2} \{S(f,t) + M(f-\varphi, t-\tau)\}, \tag{2}$$

where $\bigvee$ is the max operation [36]. Figure 10d shows an example of a dilated mel spectrogram, where we see fewer blank spaces between the distinct sounds. The amplitude across the spectrogram is increased, as seen by the colour scale. Where erosion removed the lower amplitude frequencies in the upper half of the image, dilation has restored them, treating the small areas of signal as important parts of the spectrogram that may be missing.

Erosion and dilation can be applied to the image multiple times and in combination. Erosion

and dilation are not commutative, and neither process can be reversed. The MF function of **opening** applies erosion followed by dilation. This function is commonly used to remove noise in spectrograms [39]. The process removes unwanted small regions of noise and enhances the shapes that remain, acting as a smoothing noise removal filter. The opening operation, $\circ$, is defined by

$$S \circ M = (S \ominus M) \oplus M, \tag{3}$$

where M is a fixed SE [36]. Figure 10e shows an example of an opened mel spectrogram, where the erosion step has removed some of the less prominent peaks and noise at low amplitudes. Then the dilation step is applied, which has enhanced the remaining frequencies. We see that this has combined the results shown in Figures 10c and 10d in removing the frequencies between 1024 and 2048 Hz, and enhancing the areas of higher amplitude.

Closing performs dilation then erosion on the spectrogram, first filling small gaps then smoothing the boundary. This function is also used for noise removal and smoothing rough boundaries. Closing, $\bullet$, is defined by

$$S \bullet M = (S \oplus M) \ominus M, \tag{4}$$

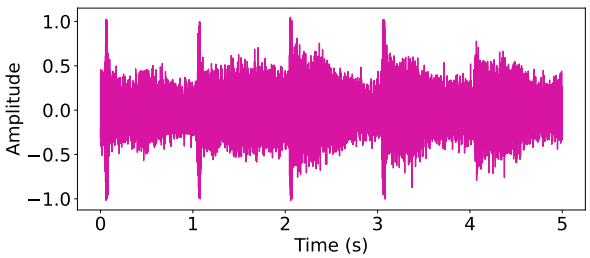
where M is a fixed SE [36]. Figure 10f shows an example of a closed mel spectrogram. This restores the smaller areas with lower amplitude, such as the high frequency peaks, as well as further enhancing the prominent lower frequency signal. Erosion then removes some of the signal to define the boundaries of the spectrogram shape. This effect can be seen when comparing Figures 10d and 10f, as the spectrogram appears to have more definition in the structure of the shape once erosion has been applied to the dilated image.

The choice of using opening or closing depends on the properties of the image. Closing may be preferred if there are missing data points or gaps that need to be filled, as opening removes smaller details first. However, opening is chosen in situations where there is more noise to be removed, and the structure of larger objects needs to be preserved.

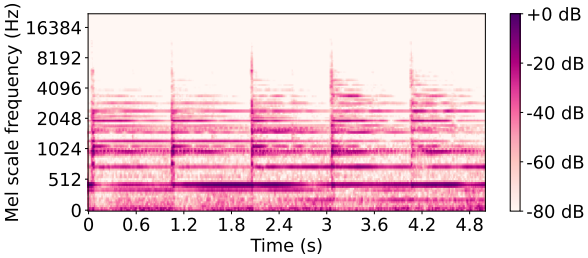
4.1.2 Results

We apply the four morphological filtering functions to the demonstrative pipelines in order to explore how they affect the classification performance. The FSDD dataset is used to first evaluate how the methods perform on a clean set of sound recordings with minimal noise. From this we can begin to understand the key differences between erosion, dilation, opening and closing, and gather insights on why the order of operations may be important for classification.

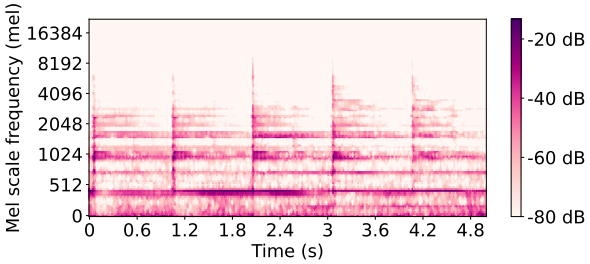
For the spoken digit classification task, we use our default SE of a disk with radius 1. After generating the spectrograms for all sound recordings, each MF operation is applied, creating



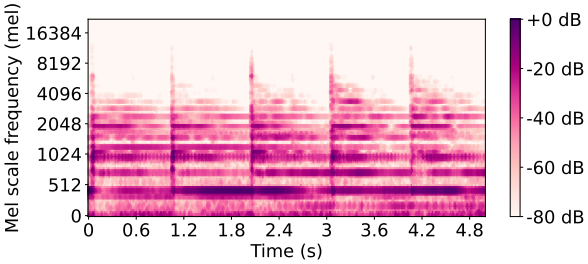
(a) Time waveform of church bells sound clip



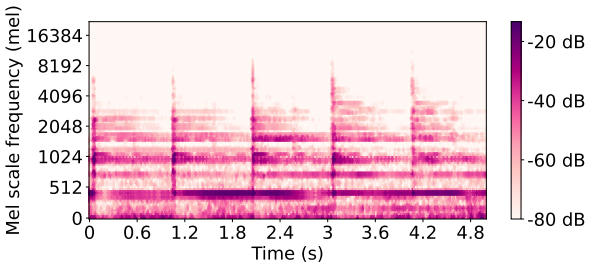
(b) Mel spectrogram of the church bells clip



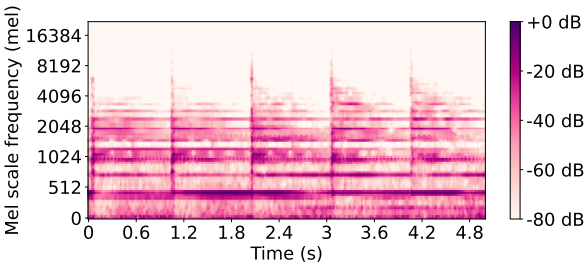
(c) Eroded mel spectrogram



(d) Dilated mel spectrogram



(e) Opened mel spectrogram



(f) Closed mel spectrogram

Figure 10: Morphological filtering functions applied to a sample audio file using structuring element of disk with radius 1. (a) Time waveform of the signal. (b) Mel spectrogram of the signal. Resulting mel spectrograms after applying erosion (c), dilation (d), opening (e) and closing (f).

four sets of altered images. We use these as input to our CNN defined in Chapter 3. Table 3 shows the results of the pipelines with MF functions applied to each dataset. Figure 14a indicates that erosion and dilation were the worst performing operations. Opening and closing have the highest F1 scores of 88.98% and 91.24% on this dataset, showing that the combination of dilation and erosion operations is beneficial for this task. We see that closing was better than opening, additionally highlighting the importance of the order of operations. In this case, dilation before erosion means that gaps in the structure of the shapes and objects were first filled in and connected, making the features more prominent and discriminative. Opening may have removed crucial details during the first erosion step, erroneously removing features that were mistaken for noise. For this application, the results show that overall, the spectrograms benefitted from being enhanced rather than reduced. This result aligns with our knowledge of the FSDD data as we know that there is already minimal noise in the original recordings. Hence, the closing function was best suited for this task.

We apply the same process to the ESC-50 and BBD datasets, which we know contain more

noisy sound clips. The results from the model are shown in Figures 14b and 14c . We see that the performance has slightly improved from the initial results using the raw data spectrograms. However, the model still performs poorly, with the highest F1 score being 33.68% for closing. Similarly for the BBD dataset, the results improved by a maximum of 4.77% by applying dilation. There are several factors that contribute to the lack of improvement in the results. There may have been too much lost information after applying the filters. The shape of the spectrograms is changed, which alters its structure. Although some noise is reduced, there could also be loss of important information that is required for classification.

	Baseline	**Erosion**	**Dilation**	**Opening**	**Closing**
FSDD	0.9133	0.8612	0.8660	0.8898	0.9124
ESC-50	0.2851	0.2867	0.3248	0.3050	0.3368
BBD	0.0818	0.0908	0.1295	0.0957	0.1154

Table 3: F1 scores of morphological filtering operations applied to each dataset for the spectrogram pipeline, compared to the baseline result.

With morphological filtering, there is a risk of over-smoothing the spectrogram, depending on the choice of structuring element. During dilation, the important details may be blurred out, as seen in Figure 10d where each frequency is less defined. Conversely, over-segmentation can occur, where the spectrogram is split into smaller, less coherent regions. This can make the shapes and patterns less recognisable as too much of the signal is eroded or removed.

One of the significant attributes of MF is that each pixel is treated independently, based on the local neighbourhood of pixels defined by the structuring element. The filters may have performed poorly because the global context of the image is not considered. For classification based on spectrogram images, it is important to understand the relationships between the different frequencies and time frames that define the audio, and this may not be captured through morphological filtering [40].

These results suggest that morphological filtering as a standalone method is not sufficient for effective feature extraction, but may be beneficial in combination with other methods such as MFCC extraction.

4.2 Combining morphological filtering with MFCC extraction

We next extend the pipeline to combine MF with MFCC extraction. Figure 11 illustrates the process, where in step 2, we generate the spectrograms and apply the filtering functions. Then in step 3, these morphologically filtered spectrograms are converted into MFCCs for the machine learning step. The difference here is that the audio or images do not need to be resized, so we can apply the operations on the full spectrograms instead of shortened ones. The intuition of this is to retain more information through MFCC extraction than with cut spectrogram images, while using the spectrograms in the preprocessing stage.

The resulting performance metrics (F1 scores) are shown in Figure 15 (see Table 4 for results

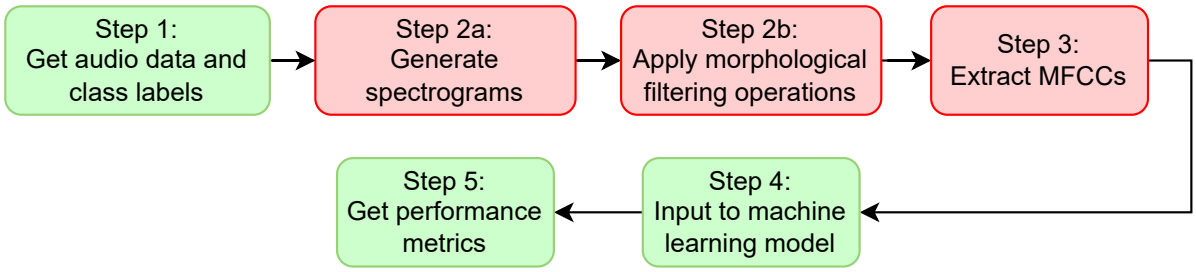


Figure 11: Adapted pipeline for combining MFCC extraction with morphological filtering.

table). We observe that representing the morphologically filtered signals as MFCCs improves the performance for some cases. This is due to the main benefits of extracting coefficients of equal length rather than using images of equal size, as all of the information in the signal is considered during MFCC extraction.

	Baseline	**Erosion**	**Dilation**	**Opening**	**Closing**
FSDD	0.9512	0.7545	0.9099	0.9012	0.9250
ESC-50	0.4080	0.3376	0.4293	0.3874	0.4304
BBD	0.2092	0.1368	0.2154	0.2096	0.2574

Table 4: F1 scores of morphological filtering operations applied to each dataset for the MFCC pipeline, compared to the baseline results.

For the FSDD, we see that closing is the best performing MF operation, and improved from the corresponding operation in the spectrogram pipeline. Dilation and opening show the highest improvement, suggesting that the spectrograms may contain specific details that could be easily ignored through erosion. This is also evidenced by the decrease in performance for the erosion operation, where the F1 score falls by 10.67% from the baseline. The sound clips in the FSDD dataset are short and clipped to contain the main spoken signal during preprocessing. Since the signals here contain the important information already, erosion only reduces the amount of useful information given to the model.

The results for the ESC-50 dataset show an improvement for all operations compared to the spectrogram pipeline. Dilation and closing have an improved F1 score compared to the baseline MFCC extraction method. The closing function involves smoothing the spectrogram and compensating for small irregularities and gaps. This was the best performing MF operation with an F1 score of 43.04%, suggesting that the data has more intricate variations and closing may have helped to enhance the signal-to-noise ratio. In doing this, the relevant features in the data are prevented from being eroded, due to the dilation step occurring first. This justifies the results for dilation performing well, but not as effectively as closing. The benefits of enhancing the smaller details using dilation are represented by the increase in F1 score by over 2% from the baseline, however we see that the subsequent erosion step is required to further improve the result. Dilation is able to emphasise the important parts of the signal, but this is paired with introduced noise, which closing is able to handle. The F1 score for erosion further validates the insights we gain about the importance of understanding the dataset. Since the performance worsened by over 7% from the initial result, this indicates that the ESC-50 data benefits from

signal enhancement. The erosion operation is useful when combined with dilation during closing, however erosion alone removes too many of the features that in fact need strengthening.

When applied to the BBD data, we see the same behaviour in performance, with closing being the best and erosion being the poorest. Unlike the FSDD results, the MF and MFCC performance has improved from the initial spectrogram F1 scores, and all but erosion have surpassed the initial MFCC results. A possible reason for this is that the FSDD showed high performance initially, leaving little room for improvement and high chances of reducing performance due to the nature of the dataset. The BBD results suggest that for the noisier data with longer field recordings, it is beneficial to combine feature extraction methods.

These results indicate the importance of understanding the characteristics of the dataset when selecting the optimal method. It may be difficult to do this for classifying different types of sound as in the ESC-50 dataset. The BBD results display the most improvement, which show that the use of MFCCs with MF operations can be effective in handling this type of data with a lower signal-to-noise ratio. From this, we learn it is key to ensure the methods used are well suited to the types of sound that are being classified, taking domain-specific knowledge into account. It is possible to further investigate the aspect of noise removal, particularly for the ESC-50 and BBD datasets. In Chapter 4.3, we evaluate the effects of incorporating spectral subtraction into the MFCC pipeline.

4.3 Spectral subtraction with MFCCs and morphological operations

Spectral subtraction is an audio processing technique used for noise reduction and signal restoration via subtraction of the estimated noise [34]. The noise spectrum can be estimated in a variety of ways, depending on the nature of the data. It is commonly assumed the noise is stationary, meaning the noise spectrum does not vary significantly over time, an assumption we make here. The noise estimation is then created by taking the mean of the spectrogram. This assumption makes the process of estimating noise more computationally efficient and is useful for real-time applications such as Merlin. Taking the mean of the spectrogram provides a tentative approach to estimation, reducing the possibility of overestimating the noise spectrum and misclassifying the signal as noise.

In Chapter 4.2, we found that the MFCC pipeline combined with morphological filtering operations performed well compared to using spectrogram images. We therefore investigate this pipeline further by applying spectral subtraction. The pipeline is adapted as shown in Figure 12. Steps 2c and 2d are added to the preprocessing stage, where we estimate the noise of each spectrogram and remove it from the image before extracting the MFCCs.

For the original spectrum of the sound, $X(f,t)$, we denote the estimated noise spectrum as $N(f,t)$. The processed spectrogram after applying spectral subtraction, $X_{SS}(f,t)$ is given by

$$X_{SS}(f,t) = X(f,t) - \alpha N(f,t), \tag{5}$$

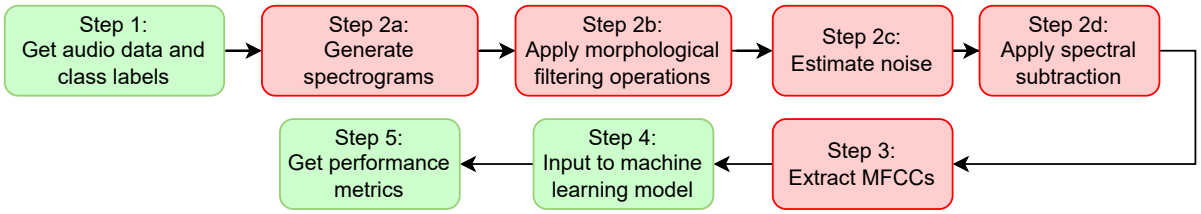


Figure 12: Classification pipeline for MFCC extraction extended with morphological filtering and spectral subtraction.

where the parameter $\alpha \in [0, 1]$ allows us to incorporate different levels of importance of the noise estimate. In our initial investigation, we arbitrarily set $\alpha = 0.5$ as a baseline to compare the performance of the pipeline with spectral subtraction. This is a parameter which can be chosen to optimise performance, as discussed in Chapter 6.1.

To visualise the implementation of spectral subtraction, we return to the example from Chapter 4.1. We estimate the noise from the spectrogram of the church bells sound, generated by computing the mean at each mel band along the time axis. This is subtracted from the original image, resulting in the spectrogram shown in Figure 13. We observe that spectral subtraction performs a thresholding effect on the spectrogram, where the portions of audio with higher amplitude are enhanced from the lower amplitude background noise.

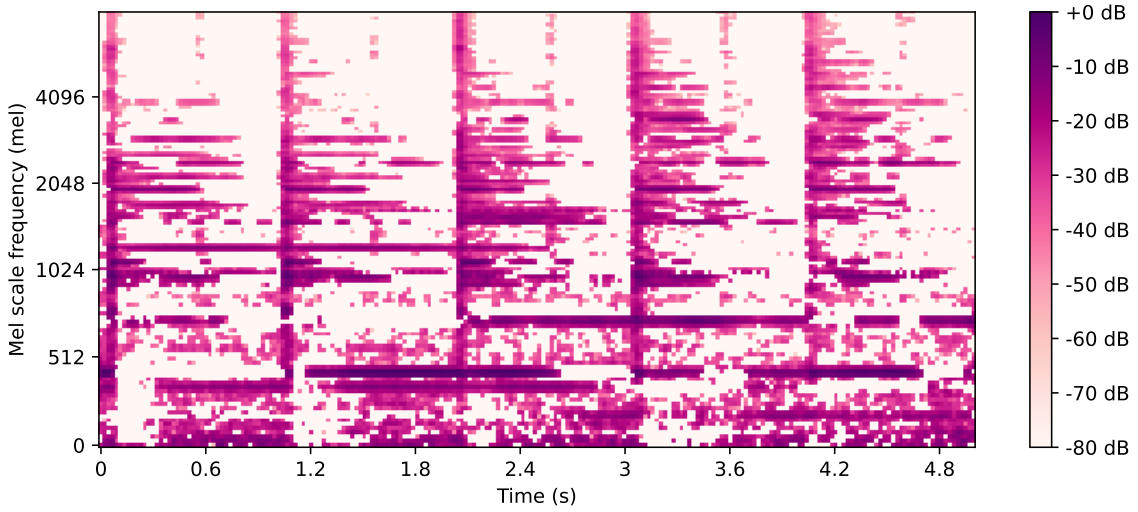


Figure 13: Mel spectrogram of the church bells sound from ESC-50 with spectral subtraction applied ($\alpha = 0.5$).

The resulting performance metrics of the MFCC pipeline with morphological filtering and spectral subtraction are shown in Table 5 and illustrated in Figure 16. Similarly to the morphological filtering pipeline, we see that erosion continues to be the worst performing method. This could be because spectral subtraction paired with erosion applies a far too aggressive reduction to the spectrograms. The most detrimental decrease in performance is seen in for the FSDD, where the F1 score with erosion is 15.84% lower than the next best combination of methods. The improvement between applying morphological filtering alone and with spectral subtraction is minimal for erosion. This shows that the addition of spectral subtraction is not always benefi-

cial and the effects of erosion may be similar to that of spectral subtraction, resulting in minimal improvements.

	Baseline	**Erosion**	**Dilation**	**Opening**	**Closing**
FSDD	0.9512	0.7595	0.9261	0.9179	0.9566
ESC-50	0.4080	0.4007	0.4282	0.3978	0.4600
BBD	0.2092	0.1572	0.2099	0.2264	0.2601

Table 5: F1 scores of morphological filtering operations and spectral subtraction applied to each dataset for the MFCC pipeline.

From Figure 16, we also observe that spectral subtraction with closing is the best performing combination of techniques across all datasets. The ESC-50 dataset has the largest amount of improvement in the F1 score compared with the MFCC baseline, highlighting that additional data processing techniques are essential in feature extraction of field recordings. The BBD results show overall improvements of 12.34% and 17.83% from the initial MFCC and spectrogram pipelines respectively. As found in Chapter 4.2, closing gave the best F1 scores for all datasets, however this only improved from the baseline for the BBD dataset. With the application of spectral subtraction, we see that the scores have exceeded the baseline for all datasets, showing its potential.

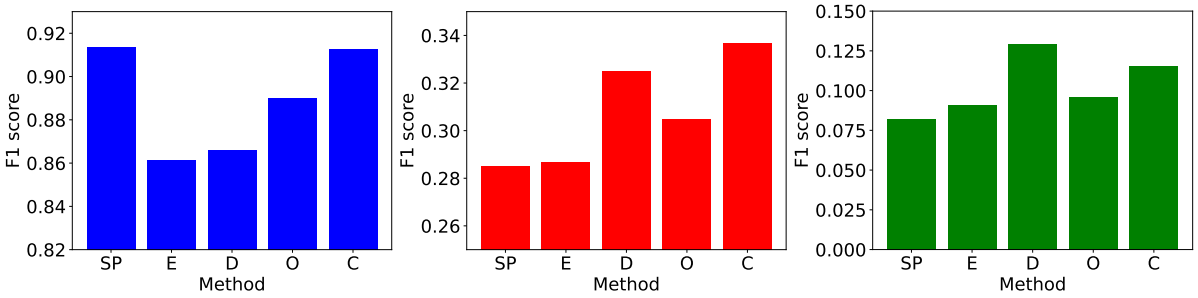
For the BBD dataset, the results show that all spectral subtraction methods apart from erosion improved the baseline classification performance. This indicates that spectral subtraction works better for data with longer clips like in the BBD, where some recordings have a high background-to-sound ratio. When there is more background than sound, the noise estimated during spectral subtraction is more representative of the actual noise. Therefore, this improves the task of highlighting the most significant audio segments. This is further explained by the smaller differences in F1 scores for the FSDD dataset. In addition to there being less room for improvement, the FSDD data is cleaner and contains minimal portions where the sound is absent. Spectral subtraction bases the estimates of noise on these areas, therefore if this is already minimised, the effects are not as advantageous. During spectral subtraction for sounds with minimal noise, the estimates will incorporate parts of the main sound, resulting in a less accurate representation and poorer performance. This is evidenced by the FSDD results struggling to overtake the baseline, and only increasing minimally if successful.

We are aware that the α parameter for the spectral subtraction method is chosen to be $\alpha = 0.5$ for this part of the investigation. The value will depend on the level of noise in the recordings, and how much reduction we believe the spectra need, considering the choice of MF operations or other processing techniques. Depending on the application, it could be the case that a different value of α will give the best result for each dataset or even recording. This optimisation task is another challenge in sound classification, as it can quickly become time consuming and computationally expensive to tailor the process to each individual case. In Chapter 6, we perform an initial investigation into the effect of the parameter on each dataset as a whole, to understand the general considerations that may arise in choosing their values.

5 Results

5.1 Spectrogram pipeline with morphological filtering

After applying each of the four MF operations to the spectrograms, the F1 scores are as shown in Figure 14. Compared to the baseline result, we see improvements for the ESC-50 and BBD datasets. The results for the FSDD and ESC-50 datasets show that erosion is the worst performing function and closing is the best performing for these cases. For the BBD dataset, dilation is the best performing MF operation for the spectrogram pipeline.

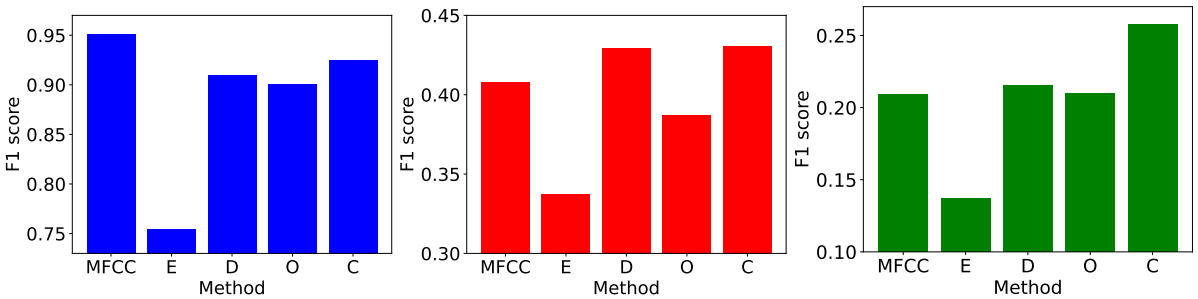


(a) FSDD (b) ESC-50 (c) BBD

Figure 14: F1 scores of morphological filtering operations, erosion (E), dilation (D), opening (O) and closing (C), applied to each dataset for the spectrogram pipeline, compared to the baseline result (SP). Results are shown for the (a) FSDD, (b) ESC-50 and (c) BBD datasets.

5.2 MFCC pipeline with morphological filtering

Figure 15 illustrates the performance of the MF operations applied to the MFCC pipeline. All datasets show that erosion performs the worst, especially for the FSDD, where there is a significant reduction in F1 score from the baseline. Closing is the MF operation that yields the highest F1 score, however the results improve from the baseline for the ESC-50 and BBD datasets only.

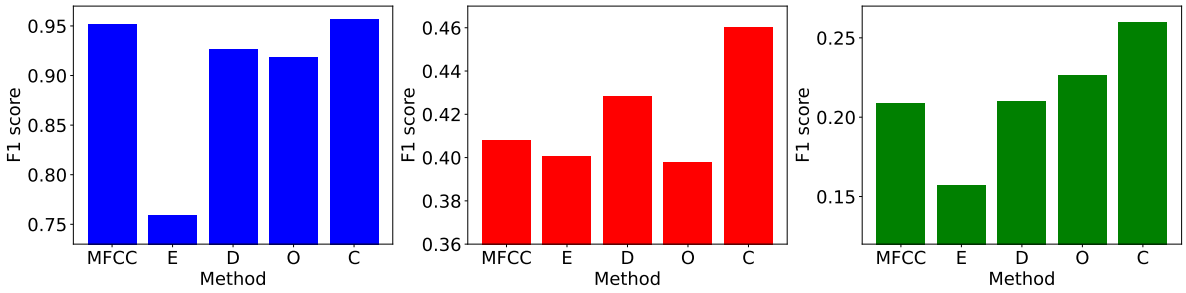


(a) FSDD (b) ESC-50 (c) BBD

Figure 15: F1 scores of morphological filtering operations, erosion (E), dilation (D), opening (O) and closing (C), applied to each dataset for the MFCC pipeline, compared to the baseline result (MFCC). Results are shown for the (a) FSDD, (b) ESC-50 and (c) BBD datasets.

5.3 MFCC pipeline with morphological filtering and spectral subtraction

With the extension of spectral subtraction applied to the MFCC pipeline with MF, we obtain the results shown in Figure 16. We see that closing is the only method that is able to surpass the MFCC baseline for the FSDD dataset, and this MF operation is the best performing for all datasets. Erosion continues to be the worst performing operation indicating that combined with spectral subtraction, this provides too much of an alteration to the data.



(a) FSDD (b) ESC-50 (c) BBD

Figure 16: F1 scores of morphological filtering operations, erosion (E), dilation (D), opening (O) and closing (C), applied to each dataset for the MFCC pipeline with spectral subtraction, compared to the baseline result (MFCC). Results are shown for the (a) FSDD, (b) ESC-50 and (c) BBD datasets.

5.4 Summary of results

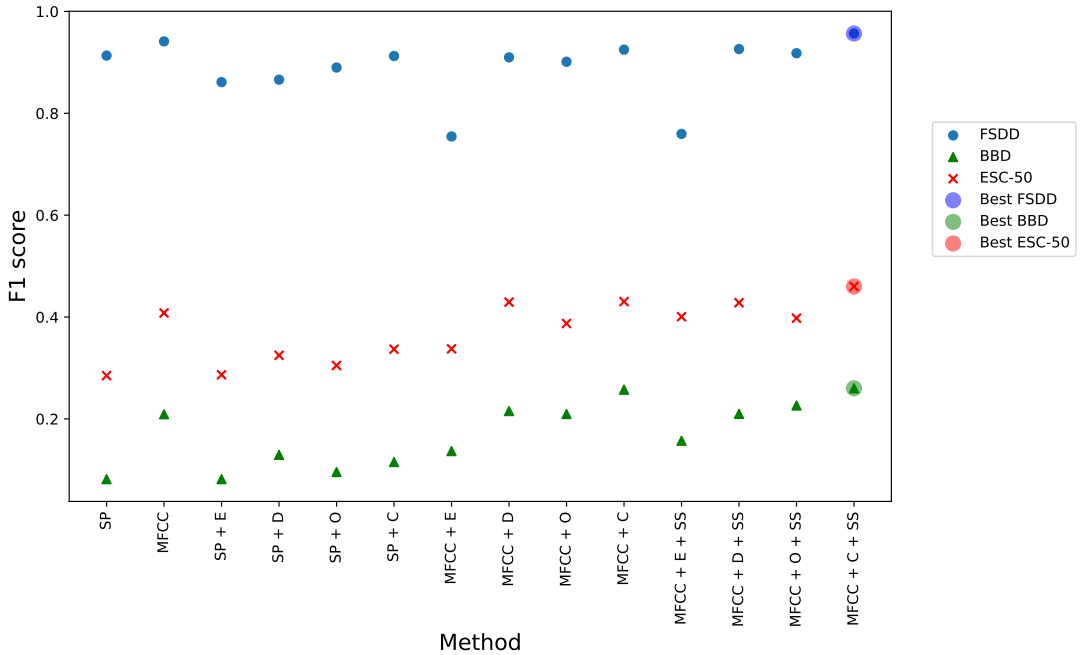


Figure 17: F1 scores for all investigated methods and datasets including the initial spectrogram pipeline (SP), initial MFCC pipeline, and extended pipelines with combinations of erosion (E), dilation (D), opening (O), closing (C) and spectral subtraction (SS) applied.

Figure 17 summarises the F1 scores for all investigated methods and processing techniques from

Chapters 3 and 4, highlighting the best performing combinations for each dataset. These will be used to further investigate the method parameters in Chapter 6. We find that the MFCC pipeline extended with closing and spectral subtraction is the best performing method across all datasets. This shows the benefits of applying the data processing techniques to the audio.

6 Parameter sensitivity

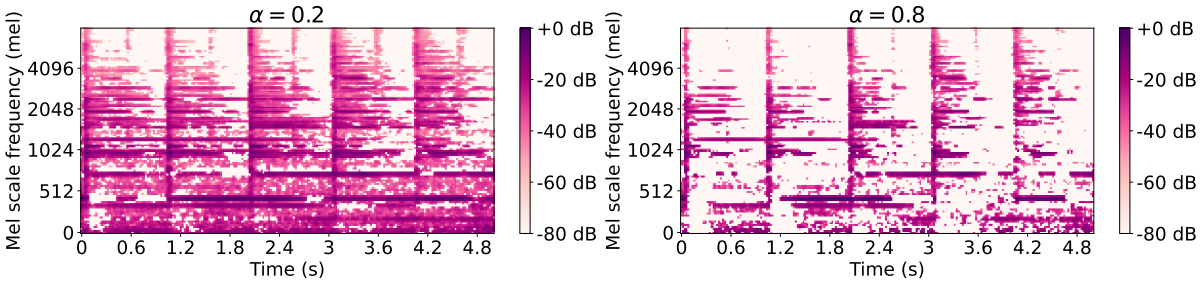
As shown in Figure 17, we find that the best performing pipeline for all datasets is MFCCs with spectral subtraction and closing. Therefore, we will use this to investigate the effect of changing some of the key parameters available in the methods. Two interesting parameters from the data manipulation investigation are the choices of SE for morphological filtering, and α for spectral subtraction. The SE determines the shape and size of the filtering element for the MF operations, while the α parameter controls the amount of noise spectrum estimation to remove. In Chapter 4, we use arbitrary values for the parameters. However, we can investigate these to understand their impacts for each application. In Chapter 6.1, we look at the effect of changing α, and in Chapter 6.2, we discuss the shape and size of the structuring element.

In Chapter 3.4, we also acknowledge the choice of the n_{MFCC} parameter for the amount of MFCCs to generate. Throughout the experimentation so far, we used the maximum value of $n_{\text{MFCC}} = 40$. For the MFCC pipelines, the number of coefficients is the size of the input for the CNN. Therefore, larger values will lead to larger inputs and potentially increase computation requirements with few benefits gained in performance. Hence in Chapter 6.3, we analyse the response to various numbers of MFCCs used to represent the audio to understand how this may differ depending on the dataset.

6.1 Spectral subtraction parameter

In Chapter 4.3, Equation (5) defines the spectral subtraction operation on the spectrum of an audio signal. We chose to include the parameter α as a means to control the amount of impact of the noise spectrum on the processed spectrogram. This choice of α can significantly alter the image. Again, we take the 'church bells' example from the ESC-50 dataset and apply spectral subtraction for $\alpha = 0.2$, 0.5 and 0.8. Figure 18 shows the resulting spectrograms for $\alpha = 0.2$ and 0.8, which we compare to Figure 13 for $\alpha = 0.5$. We see that for higher values of α, less prominent signal components remain and the signal reduction is more aggressive. Therefore, it is important that we consider how much of the signal we believe is noise and how much we wish to retain or remove. The noise signal is an estimate, meaning it will not be a completely true representation of all of the noise in the spectrum and can mislabel the main sound. There could be a higher possibility of erroneously taking away too much of the signal if α is set too high. We also make the assumption that the noise is stationary, however this may not be true in all cases, particularly for outdoor recordings where there could be a variety of background sounds contributing to the noise.

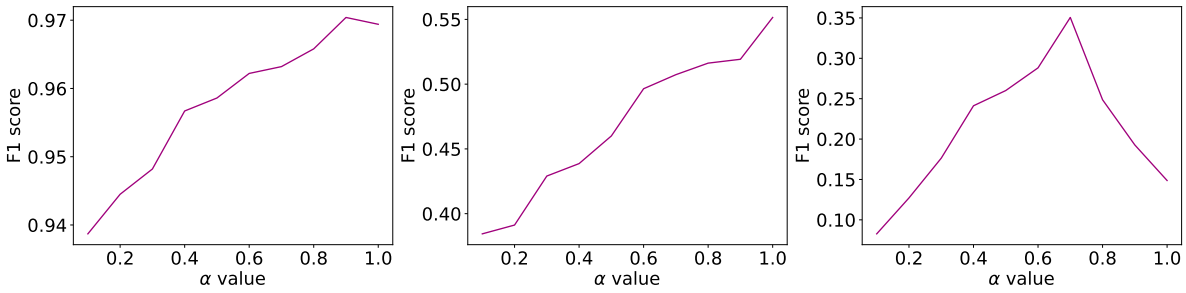
Finding the optimal value of α involves understanding the data and its applications, which can make the process more difficult. As a preliminary investigation to further work, we apply spectral subtraction for varying α values on the best performing pipeline to gain an initial understanding of how the value can affect the model performance. For simplicity of implementation, we use the same value of α across all sound files in each dataset. However, it is likely that each type of



(a) $X_{SS}(f,t)$ for $\alpha = 0.2$ (b) $X_{SS}(f,t)$ for $\alpha = 0.8$

Figure 18: Mel spectrograms for the church bells sound with spectral subtraction applied using (a) $\alpha = 0.2$ and (b) $\alpha = 0.8$.

sound within the data, such as ESC-50, will benefit from different levels of reduction, as there are a mixture of cleaner indoor sounds and noisier field recordings. The challenge lies in devising the most suitable method to determine the optimal α values for each dataset and the individual recordings. We apply spectral subtraction, for α ranging from 0.1 to 1, to each dataset and obtain the performance metrics shown in Figure 19.



(a) FSDD (b) ESC-50 (c) BBD

Figure 19: F1 scores for the MFCC pipeline with closing and spectral subtraction for varying α for the (a) FSDD, (b) ESC-50 and (c) BBD datasets.

We see that depending on the dataset, the optimal value of α changes. For all datasets, low values of α show the poorest performance, confirming the results we obtain in Chapter 4.3, where we find that the addition of spectral subtraction is beneficial. This demonstrates that thresholding the spectrograms to improve the signal-to-noise ratio aids the model in identifying the prominent frequencies within the signal during classification. For the FSDD and ESC-50 recordings, the performance increases by 3.17% and 9.15% from the baseline results, with optimal α values of 0.9 and 1.0 respectively. This could be explained by the fact that both datasets contain sound clips that initially contained minimal background noise. The FSDD clips are all short and cleaned, while the ESC-50 dataset contains some indoor sounds, which are subject to less background noise.

On the other hand, the audio in the BBD are all outdoor field recordings of varying length, where the subject may only appear in a small segment of the clip. Figure 19c shows that the peak α value occurs at $\alpha = 0.7$ and declines beyond this value. This result indicates that a more cautious approach into noise reduction must be taken for this dataset, as there is a risk

of losing essential signal information. This aligns with our knowledge of the datasets, as the identifiable sounds are more likely to be hidden in the background noise. The noise estimate is more likely to include quieter but important parts of the signal and it becomes much more difficult to distinguish from the true noise.

6.2 Changing the structuring element

In Chapter 4.1, we acknowledge that morphological filtering operations can be performed using a variety of structuring elements. Throughout the evaluation of methods, we kept the shape to be a default disk of radius 1, as shown in Figure 9a. However, the shape and size of the SE is a determining factor in the performance of the classification. Finding the optimal choice of SE is a challenging task, with little in the way of guidelines due to the vast applications of MF [41]. As discovered for the choice of α for spectral subtraction, it is expected that choice of the best structuring element will also depend on the dataset and each type of sound. We would expect that the FSDD, with the most consistent recordings of one type, would fair differently to ESC-50 which contains 50 different classes of sound. Additionally, the characteristics of the spectrograms from the BBD field recordings are much different to those in the FSDD and ESC-50. Hence, it is important that we consider this in our investigation.

We carry out an initial exploration in altering the SE from a disk with radius 1 (disk 1). We consider the star shape and a larger radius of 3, combining these parameters and applying them to the MFCC pipeline with spectral subtraction ($\alpha = 0.5$). The resulting metrics are shown in Figure 20, with a summary of the best performing combinations displayed in Table 6.

	FSDD	**ESC-50**	**BBD**
Shape	Disk	Star	Star
Radius	1	1	1

Table 6: Summary of best performing SE shape and radius for each dataset.

We see that for the FSDD dataset, our initial choice of disk 1 is the best performing SE and no alternative elements improved the result for each MF operation with spectral subtraction. Star 3 results in the poorest performance overall, followed by disk 3. For the disk shape, the largest drop in performance between radius 1 and 3 happens for erosion and opening, with a reduction in F1 score by 54.64% and 56.02% respectively. We know that the first step of opening is the erosion operation, indicating that this operation is the most influenced by the size of the SE. This result is supported by the previous results illustrated in Figure 17, where we see that the erosion and opening operations led to lower F1 scores than dilation and closing for this dataset. Using a bigger disk size takes more pixels into account when performing the MF operations. Since the FSDD contains short sounds and small spectrograms, a larger SE will alter the image more significantly than for larger images, providing a more drastic filter. As evidenced by the results, this is not useful for this dataset as we want to retain the intricacies in the spectrograms.

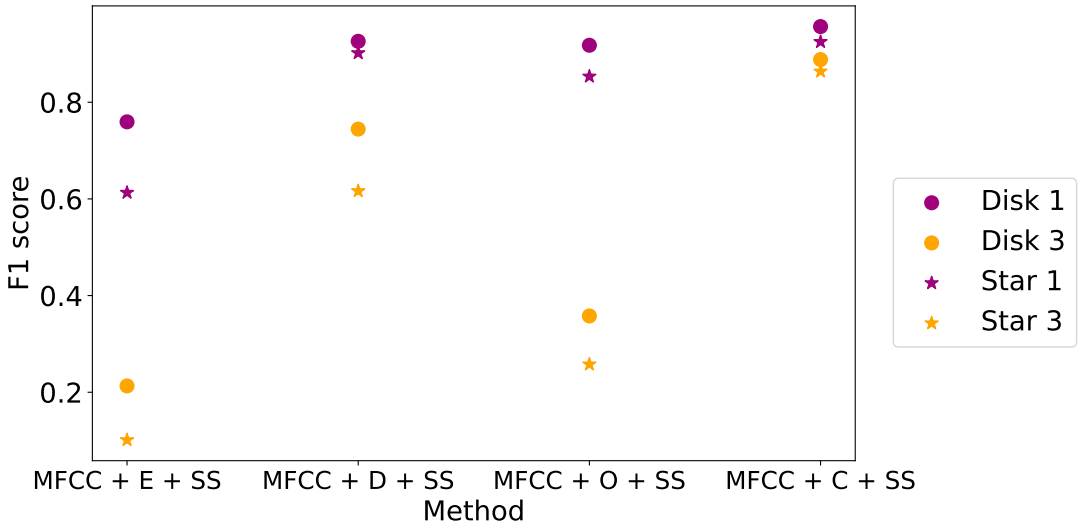
For the star shaped SEs, the results show that all stars performed worse than the disk of the

same size for the FSDD dataset. With dilation and closing, the reduction in F1 score between disk 1 and star 1 was 2.43% and 3.15% respectively. This suggests that, for a smaller radius, the shape has low impact on the F1 score. The star shape performs worse for a larger radius as the shapes are more distinct, and less relevant information is used to determine the value of the considered pixel.

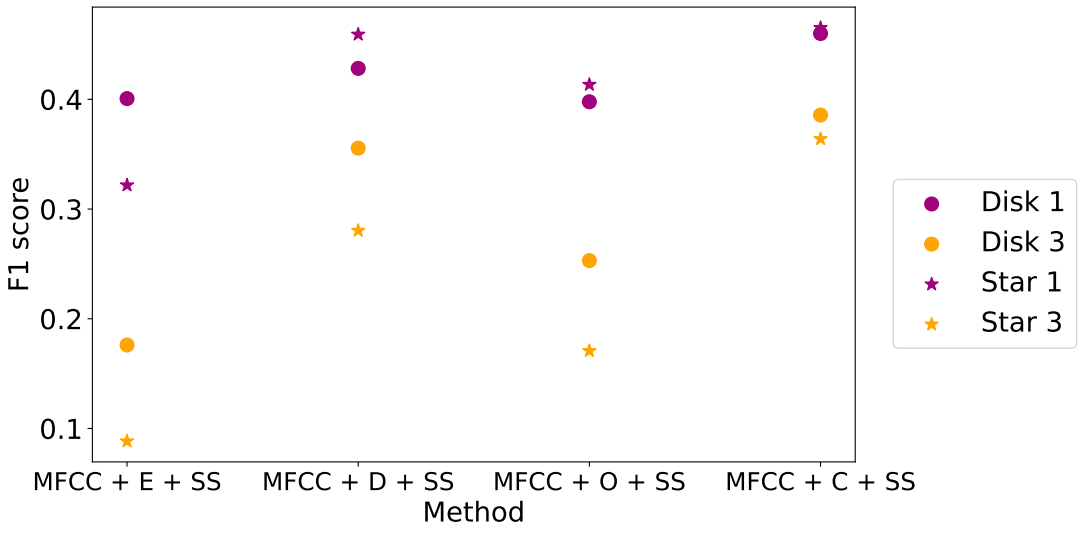
For the ESC-50 dataset, Figure 20b shows that overall, star 1 gives the best F1 score for closing. However, this is only a minimal increase 0.52% compared to the disk 1 score. The performance of the disk SE has similar behaviour to the FSDD results, where we see that a bigger disk size reduces the performance and there is too drastic of a change to identify the classes well. There is a smaller difference in performance for dilation and closing compared to erosion and opening, suggesting that closing is a more consistently performing method. This MF operation is less prone to significant changes in performance in the event of altering other parameters in the pipeline. As seen with the FSDD, the methods involving erosion as the initial MF step are more sensitive to parameter change, which could be due to the large number of classes (50). The finer details of the spectrograms are important for classification and need to be enhanced from the background.

When changing the SE to the star shape, this improved the F1 scores for all MF operations apart from erosion for the smaller radius. This shows that for the ESC-50 dataset, considering pixels diagonal to the centre pixel is important for enhancement and retaining the useful information that is otherwise ignored when using disk 1. However, when using a larger SE, the disk is still the better performing SE. The patterns in the spectrograms are not as well suited to the larger star, as we begin to incorporate more irrelevant components in the MF operations.

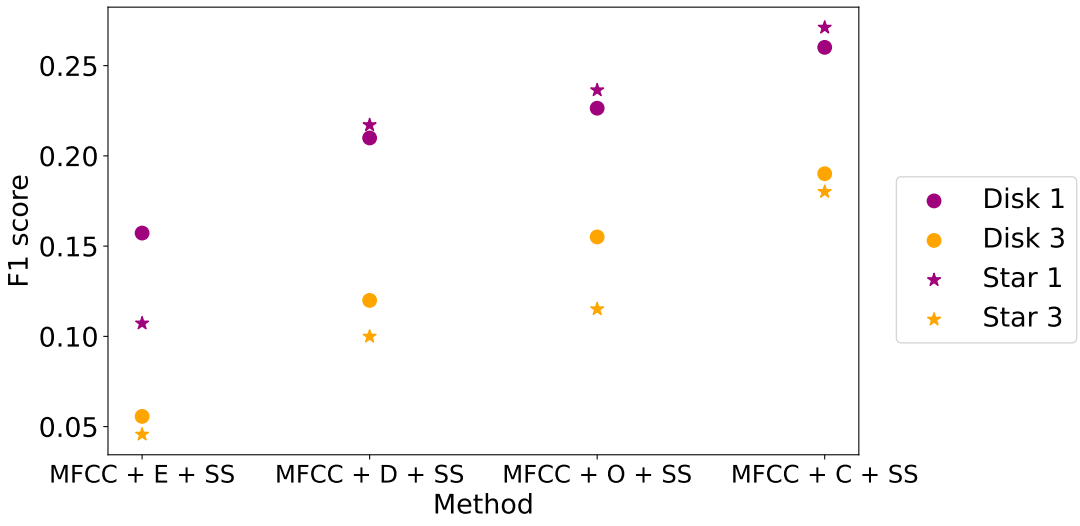
Figure 20c shows the results for the BBD dataset. Overall, we see similar results as with the ESC-50 dataset. This is likely due to the similarity in the type of data in both datasets, compared to the FSDD. We see the F1 score improves by 1.1% from the best performing BBD method in Figure 17. Again, this suggests that for datasets which contain more noisy and less refined sound clips, the star shape allows for diagonal components in the spectrograms to be considered which proves beneficial on a smaller scale. We notice that the performance of the erosion method does not improve when the SE is altered in shape or size from the default disk 1, further confirming that erosion should be used with caution as the type of SE is more influential on the performance with this MF operation.



(a) FSDD



(b) ESC-50



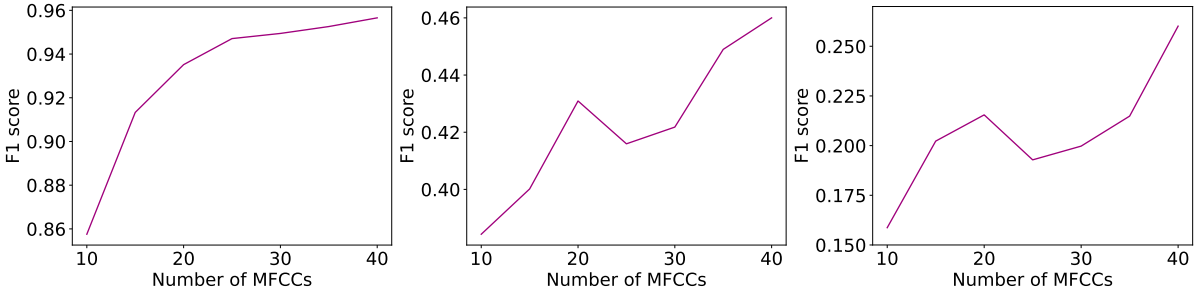
(c) BBD

Figure 20: F1 scores for the MFCC pipeline with morphological filtering and spectral subtraction, with different structuring elements for the (a) FSDD, (b) ESC-50 and (c) BBD datasets.

6.3 Investigating the $n_{\mathbf{MFCC}}$ parameter

A fundamental parameter for the MFCC pipeline is the number of features we choose to extract, with a possible maximum of $n_{\text{MFCC}} = 40$. This determines the size of the input to the CNN. Therefore, the more features we extract, the more computation is required by the model. However, representing the audio by more features allows for more information to be captured and could be beneficial for the classification task. In particular, for the application of bird species identification, a larger MFCC feature vector could provide the model with the more subtle details required for accurate prediction.

We investigate the effect of using different values of n_{MFCC} for the best performing methods on each dataset. We vary the parameter between 10 and 40 in increments of 5 and observe the performance metrics. The α parameter and shape of the SE remain as their default values of $\alpha = 0.5$ and disk 1 for this investigation. The F1 scores for each dataset are shown in Figure 21.



(a) FSDD (b) ESC-50 (c) BBD

Figure 21: F1 scores for the MFCC pipeline with closing and spectral subtraction for varying n_{MFCC} for the (a) FSDD, (b) ESC-50 and (c) BBD datasets.

From the results, we see that all datasets benefit from using the largest number of MFCC features, meaning it is useful for this application to use more features and trade off some computation power. However this is case specific, as other applications in literature have found that extracting lower numbers of MFCC features have been optimal for classification performance [42].

For the FSDD dataset, Figure 21a shows that there is a steep increase in performance from the lower values of n_{MFCC}, plateauing to the optimal of 40 features. As the sound clips in this dataset are relatively short, there are fewer STFTs taken compared to the ESC-50 and BBD data. Therefore, it is vital to extract as much information from these sounds as possible, else there is a risk that the MFCC representations will be too similar between different classes.

The ESC-50 and BBD results show similar characteristics in the way the performance reaches the optimal of $n_{\text{MFCC}} = 40$. For both datasets, the F1 scores increase up to $n_{\text{MFCC}} = 20$ and slightly decrease before improving again. However, when examining the absolute values, the difference in performance is relatively small, indicating that this behaviour may be due to randomness in the data. However, we must also consider how the MFCCs are chosen, as each feature in the vector represents different information about the signal. The first coefficient represents the

average power in the spectrum, and the second approximates the broad shape from the spectral centroid [43]. The higher order coefficients represent other details of the spectrum such as pitch. As discussed in Chapter 2.4, the discrete cosine transform is applied in the final step of extracting the features, to try and decorrelate the coefficients. However, there may still be correlation and dependency between the coefficients, explaining the results in Figures 21b and 21c. The coefficients between approximately $n_{\text{MFCC}} = 20$ and 30 may have more dependence on the higher order coefficients in their ability to capture the information in the signal. Despite this variation, the final results show that these datasets all benefit from using the maximum number of MFCC features to classify the sounds.

7 Discussion

7.1 Dataset characteristics

Each of the three datasets have varying levels of similar sounds and characteristics. The FSDD is used as the most ideal case, with short, cleaned sound clips of spoken digits. With 3000 recordings and 300 instances per class, this dataset allowed for deeper understanding of the performance of each method when the quality of the data is not a prominent issue. As expected, we find that the performance on the FSDD was the highest overall. The combination of MFCCs with closing and spectral subtraction yielded the highest F1 score of 95.66%, shown in Figure 17. Upon investigation of the method parameters, the F1 score reached 97.04% with $n_{\text{MFCC}} = 40$ and $\alpha = 0.9$.

The ESC-50 dataset is used to examine how the methods perform when classifying a larger number of labels, with 50 different types of sound. During data collection, efforts were made to keep the sound events in the foreground with minimal background noise, however this is much more difficult for field recordings [25]. Therefore, this dataset serves as an intermediate case where the data contains a mixture of indoor and outdoor recordings with varying levels of background noise. There are varying levels of noise within the clips, and we conclude that the processing methods are beneficial in ensuring the sound events are enhanced, while removing as much background noise as we can. We find that the data is best classified when using MFCCs with closing and spectral subtraction, where both the n_{MFCC} and α parameters are set to their maximum values of 40 and 1 respectively. We achieve a maximum F1 score of 55.15% using these methods and parameter values, while keeping the CNN model consistent throughout. This is a significant improvement from the baseline results, where the F1 scores were 28.51% for the spectrogram pipeline and 40.80% for the MFCC pipeline before applying additional preprocessing steps.

The BBD dataset is solely comprised of field recordings of various lengths and noise levels. It is used to understand the considerations that must be made when classifying noisy sounds. This dataset is an example of the inputs for real-time sound classification applications such as Merlin ID and Shazam. With 85 species and only 3 instances for each, this data provides us with a realistic case for the multi-class problem and allows us to compare the performance of the discussed methods on a raw set of data. From the investigation, we find, as suspected, that the performance on the BBD dataset is very low, with an F1 score of 20.92% for the MFCC pipeline with no further processing. After exploring the MF operations and noise reduction techniques, we achieved an improvement in F1 score of 8.07% when using closing and spectral subtraction. Despite the increase in F1 score, the performance is still poor on this dataset. This highlights the amount of consideration that needs to be made in the other steps of the pipeline. Using field recordings introduces a variety of additional background noises, as well as the possibility of multiple identifiable sounds in the same clip. This immediately increases the difficulty of the task, evidenced by the results in this investigation.

We gain an understanding of the conditions required for improved classification performance, both for a dataset and the processing stage. One of the main differences between the FSDD, ESC-50 and BBD datasets is that the size of the audio clips in the FSDD are significantly shorter. As Figure 17 shows, the FSDD dataset performed the best in all investigations. This confirms that if the main components of the sounds can be segmented and identified from the background noise, the model is able to classify much better. This is not the case for the field recordings and proves to be a challenge that can begin to be overcome in the data processing stage.

For the scope of this investigation, we design CNNs that are useful for the demonstrative pipelines. The architecture is kept consistent throughout while the signal processing techniques are explored. However, we acknowledge that the machine learning method is a crucial part of the classification task, with several design choices, parameters and models to consider. Therefore, the conclusions we make may be limited by the choice of CNN and machine learning methods.

7.2 Morphological filtering insights

We discover the importance of understanding the dataset we are working with and the implications of using certain processing techniques. For example, we find that the erosion operation significantly hinders classification performance particularly for the FSDD dataset. We also notice that the order of MF operations matters, evidenced by the difference in performance across all datasets. When erosion is applied first, this appears to remove too much of the signal from the spectrogram, which dilation cannot restore.

Another key insight is the influence of the structural element used within morphological filtering. Depending on the size and shape of the spectrogram, the choice of SE can significantly alter the data. We see that across all datasets, increasing the size of the SE is not beneficial as there are intricate details that must be retained. A larger SE may be more useful when the shapes we wish to enhance are larger or simpler. For this investigation, and particularly for the BBD, there are many classes of sound which may have very subtle differences in their characteristics and features. Therefore we require the SE to be smaller to allow for these details to be retained. However, this is dependent on the size of the spectrogram and resolution of the image, determined by the sampling frequency and parameters such as window size for the STFT.

7.3 Spectral subtraction methods

We investigate the effects of applying spectral subtraction in addition to morphological filtering for the MFCC pipeline. For this experiment, we use the mean of the spectrogram as the noise estimation. However, this is only one method for noise estimation and there are several ways to alternatively generate the noise spectrum. A more computationally challenging approach is to identify the moments in the audio where the main sound is not present, and estimate the noise spectrum from these portions. This can give a better estimate of the noise as the main sound

event is not considered. A drawback to this method is that the task of segmenting the audio can quickly become challenging, especially for datasets such as the BBD. This method is useful for datasets with standard characteristics such as speech recordings, but there is less clarity in distinguishing the signal from the background in field recordings. In particular for bird sounds, the birds may be heard within the background noise, making this method more challenging.

One of the key limitations to the implemented method is the assumption that the noise is stationary and we use a fixed value of α across all sounds. In reality, it is likely that the noise is time-varying and its estimation should account for this. Optimising the spectral subtraction parameters is a difficult task, but can be implemented to improve performance. For example multi-band spectral subtraction is a variation of the standard method, where the subtractive parameters are adapted in time and frequency, based on the signal-to-noise ratio [44].

7.4 Real world application

This investigation highlights the challenges of real-world applications of sound classification. The BBD provides a focus on bird species identification and is used to explore how the limiting factors of tools such as Merlin ID can be overcome. One of the main strengths of their method is that expert labelling allows for the relevant portions of the spectrograms to be identified, however this can become inefficient as the number of recordings increases. Despite this, the challenge of automatically segmenting the relevant calls is equally challenging and requires sophisticated detection methods that may not replicate the accuracy of experts. Applications like these will be faced with inputs similar to those in the BBD dataset in that there are few instances per class. Real-time sound classification tools often face the challenge of having limited amounts of raw data to train their models with, especially for lesser observed species. This, in combination with the large amount of bird species, requires being able to locate and identify the exact portions of useful sound within the recordings. It is likely that there are multiple species present in the same clip, yet the label in the data is for one species. This can be a cause of confusion during training, as it is not specified which portion of the sound relates to the species in the label. The dataset can be expanded if the sounds can be separated, however this can be difficult without manual labelling by ornithology experts. This task would require verification of sounds that may be ambiguous, or where the machine learning model alone cannot distinguish different species. Additional techniques such as image detection could be explored in order to detect multiple instances of sounds, as well as recognising characteristic patterns in the spectrograms.

8 Conclusions and further work

8.1 Conclusion

In this investigation, we apply various techniques from signal and image processing to evaluate their performance on sound classification for different datasets. We explore spectrogram images and MFCC coefficients as inputs to a CNN model. These pipelines are expanded using signal and image processing techniques of morphological filtering and spectral subtraction for feature enhancement and noise reduction. We then further explore the parameters within these methods by investigating the effects of using different structural elements in morphological filtering, changing the level of noise reduction in spectral subtraction, and the number of MFCC coefficients we give to the model. For the application of bird sound identification, we find that it is essential to apply processing techniques that can distinguish the sound event from its background, and this becomes more challenging with field recordings. From our exploration, we discover that MFCC extraction with spectral subtraction and the morphological filtering operation of closing provides the most improvement from using the raw sounds. This can be further extended by considering alternative signal transforms, machine learning models and noise reduction techniques.

8.2 Gammatone cepstral coefficients

In our investigation, we found that MFCCs were useful in extracting the features of the sounds, and performed better than mel spectrograms as inputs to the CNN. Gammatone Cepstral Coefficients (GTCCs) are an alternative feature representation. Similarly to MFCCs, the coefficients are more representative of the human auditory system and are inspired by the function of the cochlea in the ear [45]. GTCCs have been found to be beneficial in speaker recognition and speech classification tasks [46]. An extension to this work could adapt these methods to different applications such as bird species identification or other fields such as health monitoring.

8.3 Wavelet transform

One of the first steps in the sound classification pipelines is to apply the Fourier transform to the audio signal, in order to obtain a representation of the sound in the frequency domain. However, there are alternative transforms such as the wavelet and Hilbert transforms that could be explored for this task. The Fourier transform is able to identify the frequencies present in the signal but not when they appear in time, hence we use the short-time Fourier transform (STFT) which involves windowing. One of the trade-offs made with the STFT is between time and frequency resolution. Using a narrow window allows for better time resolution but brings more uncertainty in frequency resolution. The wavelet transform can simultaneously capture time and frequency information based on scale and location, while maintaining high resolution in both areas [47].

8.4 Machine learning models

The scope of this work was to investigate methods to improve the data processing step of the sound classification task. An extension for further work would be to explore the machine learning step. In our case, we use a relatively simple CNN architecture for demonstrative purposes. There are a range of machine learning models that can be investigated as alternatives to CNNs. For example, the combination of CNNs and support vector machines has been found to improve on standard methods for animal sound classification [48]. Furthermore, Xie et al. explore the use different CNN-based models specifically for bird species identification and alternative data processing techniques to tackle the sizing issue for the spectrogram pipeline [49]. Tools that have been pre-trained on similar types of sounds, such as BirdNET, could also be used and explored on datasets like the BBD dataset. This stage of the pipeline has huge potential for exploration and is an area that is continually developing.

References

[1] The Cornell Lab of Ornithology. Sound ID. URL: https://tinyurl.com/2nhwnw9b. Accessed 07/10/2023.

[2] B. Hoffman and G. Van Horn. Behind the Scenes of Sound ID in Merlin. https://www.macaulaylibrary.org/2021/06/22/behind-the-scenes-of-sound-id-in-merlin/. Accessed 07/10/2023.

[3] Cornell Lab of Ornithology and NYU Music and Audio Research Laboratory. BirdVox - Machine Listening for Bird Migration Monitoring. https://wp.nyu.edu/birdvox/. Accessed 10/10/2023.

[4] K. Lisa Yang Center for Bioacoustics. BirdNET Sound ID - The easiest way to identify birds by sound. https://birdnet.cornell.edu/. Accessed 19/10/2023.

[5] N. Robertson. Evaluate Window Functions for the Discrete Fourier Transform. URL: https://www.dsprelated.com/showarticle/1211.php. Accessed 11/10/2023.

[6] S. Braun. Windows. In S. Braun, editor, *Encyclopedia of Vibration*. Elsevier, 2001. DOI: https://doi.org/10.1006/rwvb.2001.0052.

[7] S. Stevens et al. A Scale for the Measurement of the Psychological Magnitude Pitch. *The Journal of the Acoustical Society of America*, Volume 8:185–190, 01 1937. DOI: https://doi.org/10.1121/1.1915893.

[8] N. Dave. Feature extraction methods LPC, PLP and MFCC in speech recognition. *International Journal For Advance Research in Engineering And Technology*, Volume 1, 07 2013.

[9] A. Wang. An industrial strength audio search algorithm. In *Shazam*, 01 2003.

[10] K. Doshi. How does Shazam work? Music Recognition Algorithms, Fingerprinting, and Processing. URL: https://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition. Accessed 12/10/2023.

[11] C. Frank. The Machine Learning Behind Hum to Search. URL: https://blog.research.google/2020/11/the-machine-learning-behind-hum-to.html. Accessed 13/10/2023.

[12] J. Lyon. Google's Next Generation Music Recognition. URL: https://blog.research.google/2018/09/googles-next-generation-music.html. Accessed 13/10/2023.

[13] B. Arcas et al. Now playing: Continuous low-power music recognition. In *NIPS 2017 Workshop: Machine Learning on the Phone*, 2017. DOI: https://doi.org/10.48550/arXiv.1711.10958.

[14] Y. Zeinali et al. Heart sound classification using signal processing and machine learning algorithms. *J. R. Soc. Interface*, 7, 2022. DOI: https://doi.org/10.1016/j.mlwa.2021.100206.

[15] Dalal Bardou, Kun Zhang, and Sayed Mohammad Ahmad. Lung sounds classification using convolutional neural networks. *Artificial Intelligence in Medicine*, 88:58–69, 2018. DOI: https://doi.org/10.1016/j.artmed.2018.04.008.

[16] V.R. Krishnan and B. Anto. Feature parameter extraction from wavelet subband analysis for the recognition of isolated malayalam spoken words. *International Journal of Computer and Network Security*, 1(1):52–55, 2009.

[17] M. Cutajar et al. Comparative study of automatic speech recognition techniques. *IET Signal Processing*, 7(1):25–46, 2013. DOI: https://doi.org/10.1049/iet-spr.2012.0151.

[18] X. Du and P. He. The Clustering Solution of Speech Recognition Models with SOM. In *Advances in Neural Networks - ISNN 2006*, pages 150–157. Springer Berlin Heidelberg, 2006. DOI: https://doi.org/10.1007/11760023_23.

[19] D. Muller et al. A Connectionist Approach to Speech Understanding. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 3790–3797, 2006. DOI: https://doi.org/10.1109/IJCNN.2006.247398.

[20] M. Korba et al. Robust speech recognition using perceptual wavelet denoising and mel-frequency product spectrum cepstral coefficient features. *Informatica*, 32(3), 2008.

[21] K. Piczak. Environmental sound classification with convolutional neural networks. In *2015 IEEE MLSP*, pages 1–6, 2015. DOI: https://doi.org/10.1109/MLSP.2015.7324337.

[22] Z. Jackson. Free Spoken Digit Dataset (FSDD). URL: https://github.com/Jakobovski/free-spoken-digit-dataset. Accessed 19/10/2023.

[23] R. Tatman. British Birdsong Dataset. URL: https://www.kaggle.com/datasets/rtatman/british-birdsong-dataset. Accessed 19/10/2023.

[24] S. Stoudt et al. Identifying engaging bird species and traits with community science observations. *PNAS*, 119(16), 2022. DOI: https://doi.org/10.1073/pnas.2110156119.

[25] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, page 1015–1018, 2015. DOI: https://doi.org/10.1145/2733373.2806390.

[26] F. Font et al. Freesound technical demo. In *Proceedings of the 21st ACM International Conference on Multimedia*, page 411–412, 2013. DOI: https://doi.org/10.1145/2502081.2502245.

[27] B. McFee et al. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015. DOI: https://doi.org/10.25080/Majora-7b98e3ed-003.

[28] M. Massoudi et al. Urban Sound Classification using CNN. In *2021 6th International Conference on ICICT*, pages 583–589, 2021. DOI: https://doi.org/10.1109/ICICT50816.2021.935862.

[29] R. Mandelbaum. New Shazam for Birds Will Identify That Chirping for You. https://gizmodo.com/new-shazam-for-birds-will-identify-that-chirping-for-yo-1847164904Accessed 21/10/2023.

[30] L. Weiyang et al. Large-margin softmax loss for convolutional neural networks, 2017. DOI: https://doi.org/10.48550/arXiv.1612.02295.

[31] A. Wilson et al. The Marginal Value of Adaptive Gradient Methods in Machine Learning, 2018. DOI: https://doi.org/10.48550/arXiv.1705.08292.

[32] S. Sharma. Implications of Pooling Strategies in Convolutional Neural Networks: A Deep Insight. *FCDS*, 44(3):303–330, 2019. DOI: https://doi.org/10.2478/fcds-2019-0016.

[33] I. Young. Image analysis and mathematical morphology. *Cytometry*, 4:184–185, 09 1983. DOI: https://doi.org/10.1002/cyto.990040213.

[34] S. Vaseghi. *Spectral Subtraction*, pages 242–260. Vieweg+Teubner Verlag, Wiesbaden, 1996. DOI: https://doi.org/10.1007/978-3-322-92773-6_9.

[35] A. Bovik. Basic Binary Image Processing. In *The Essential Guide to Image Processing*, pages 69–96. Academic Press, Boston, 2009. DOI: https://doi.org/10.1016/B978-0-12-374457-9.00004-4.

[36] J. Cadore et al. Auditory-Inspired Morphological Processing of Speech Spectrograms: Applications in Automatic Speech Recognition and Speech Enhancement. *Cognitive Computation*, 5:426–441, 12 2013. DOI: https://doi.org/10.1007/s12559-012-9196-6.

[37] P. Maragos. Chapter 13 - morphological filtering. In *The Essential Guide to Image Processing*, pages 293–321. Academic Press, 2009. DOI: https://doi.org/10.1016/B978-0-12-374457-9.00013-5.

[38] C. Li et al. Diffraction imaging using a mathematical morphological filter with a time-varying structuring element. *GEOPHYSICS*, 86:1–51, 2021. DOI: https://doi.org/10.1190/geo2020-0177.1.

[39] N. Evans et al. Noise Compensation using Spectrogram Morphological Filtering. 09 2002.

[40] X. Wang and Z. Zhu. Context understanding in computer vision: A survey. *Computer Vision and Image Understanding*, 229:103646, 2023. DOI: https://doi.org/10.1016/j.cviu.2023.103646.

[41] S. Gautam and S. Brahma. Guidelines for selection of an optimal structuring element for mathematical morphology based tools to detect power system disturbances. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–6, 2012. DOI: https://doi.org/10.1109/PESGM.2012.6345105.

[42] L. Grama and C. Rusu. Choosing an accurate number of mel frequency cepstral coefficients for audio classification purpose. In *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, pages 225–230, 2017. DOI: https://doi.org/10.1109/ISPA.2017.8073600.

[43] D. Mitrović et al. Features for content-based audio retrieval. In *Advances in Computers: Improving the Web*, volume 78 of *Advances in Computers*, pages 71–150. Elsevier, 2010. DOI: https://doi.org/10.1016/S0065-2458(10)78003-7.

[44] N. Upadhyay and A. Karmakar. Speech Enhancement using Spectral Subtraction-type Algorithms: A Comparison and Simulation Study. *Procedia Computer Science*, 54:574–584, 2015. DOI: https://doi.org/10.1016/j.procs.2015.06.066.

[45] R.D. Patterson et al. Complex sounds and auditory images. In Y. CAZALS, K. HORNER, and L. DEMANY, editors, *Auditory Physiology and Perception*, pages 429–446. Pergamon, 1992. DOI: https://doi.org/10.1016/B978-0-08-041847-6.50054-X.

[46] X. Valero and F. Alías-Pujol. Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification. *Multimedia, IEEE Transactions on*, 14:1684–1689, 2012. DOI: https://doi.org/10.1109/TMM.2012.2199972.

[47] ML Fundamentals. A guide for using the Wavelet Transform in Machine Learning. https://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/ Accessed 15/12/2023.

[48] K. Ko et al. Convolutional feature vectors and support vector machine for animal sound classification. volume 2018, pages 376–379, 07 2018. DOI: https://doi.org/10.1109/EMBC.2018.8512408.

[49] J. Xie et al. Investigation of Different CNN-Based Models for Improved Bird Sound Classification. *IEEE Access*, 7:175353–175361, 2019. DOI: https://doi.org/10.1109/ACCESS.2019.2957572.