The BlindCamera: Perception of Object Interaction Events from Audio Sensors

Daniel Whettam

A first year summer project submitted to the University of Bristol's UKRI Centre for Doctoral Training in Interactive Artificial Intelligence

October 7, 2020

9534 words

Abstract

In this work we consider the problem of action recognition from audio alone, in the context of fine-grained events. These are actions that occur over a very short time period, such as the closing of a draw, or chopping an onion. We use the audio from the EPIC-KITCHENS dataset [6, 7], which consists of fine-grained actions in the kitchen. Using this data we train two models; a off-the-shelf ResNet50 [14], and a state-of-the-art model for audio classification on the DCASE challenge - a ResNet-based model with frequency aware convolutions and receptive field regularisation [19, 20]. Using these two models we perform an in-depth investigation into the problem of fine-grained audio-based action recognition, demonstrating the advantages and shortcomings of each approach. In particular, we consider the data representation and optimisation process for our task and carefully analyse the performance for both of the models, with the aim of determining how effective audio is as a modality for performing action recognition, as well as how well state-of-the-art audio classification approaches transfer to the fine-grained activity recognition domain.

We show that it is very possible to achieve competitive results using audio-alone for both the ResNet50 and frequency aware models, achieving 43.72% top-1 accuracy on verbs, compared to 65.26% for a model comprising of three modalities: RGB, optical flow, and audio. We also show that whilst the frequency aware model is competitive for our task, it does not perform as well as a well-tuned ResNet50, suggesting that there is more to be done to effectively transfer such a technique to the domain of fine-grained action recognition.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This work investigates the use of neural network based approaches for performing fine-grained event recognition from the audio stream alone. Since the introduction of AlexNet [21], deep learning has become an extremely effective technique for many domains and, in particular, has proven to be extremely effective for many computer vision tasks through the use of Convolutional Neural Networks [10, 22]. Most computer vision work is focused around images; common tasks include object detection, image recognition, image segmentation, and image generation. As our ability to perform these tasks improves, attention is shifting towards video instead of images. Video consists of a sequence of images, meaning many image-based techniques are applicable. However, it also presents some additional challenges. Most notably, video understanding requires some understanding of context; a single frame cannot be understand without placing it within the context of the rest of the video. Consequently, learning video representations must be done sequentially in order to capture the contextual nature of video. Another distinguishing factor between video and images is the presence of audio; audio offers a wealth of additional information that is often vital for humans to understand different scenarios. Multi-modal work (e.g. [3, 4, 17]) considers both video and audio jointly, often increasing a model's ability to learn how to perform the task. Unlike these multi-modal works, this project considers audio alone. Multi-modal work is often largely focused on the visual aspects of a video, and the audio is considered supplementary to the video stream. By considering audio independently we can understand how to most effectively utilise the audio-stream for future use in a multi-modal context. More specifically, we are focusing on classifying fine-grained events, such as placing down an object, or chopping an onion. Classifying audio in this form has received little attention in the literature and has many potential applications within the smart-home setting.

1.1 Motivation

Smart-home devices are becoming increasingly ubiquitous in people's homes, meaning many people are exposed to always-on style microphones. These devices can be very useful for organisation, performing household tasks, as well as entertainment. Whilst these devices are now excellent at recognising what we say they do not have the ability to recognise our actions within the home. Extending smart-home devices to include action recognition could greatly enhance these already very capable devices, providing capabilities such as recipe instructions that follow along according to your actions.

Action recognition of this nature could be performed through a visual, or multi-modal approach, which have both been previously explored [17, 29]. Alternatively, action recognition could be performed though an entirely audio-based approach, giving some distinct advantages. Firstly, microphones have already become commonplace in many modern homes through smart-home devices, making audio-only action recognition easier to integrate. Secondly, public perception is that microphones are a more private sensor than cameras - a Siri/Alexa equivalent that recorded your actions through a camera and provided similar smart home facilitates would be considered highly invasive. By demonstrating the utility of fine-grained action recognition, the perception that audio sensors are more private than video can be broken.

Recent work [17] has demonstrated that audio plays a significant role in performing effective action recognition. Building on this observation, we aim to perform action recognition from a simple audio signal of events happening within the home. In particular, this project is focused on action recognition from the audio signals of fine-grained events (such as opening a draw, chopping an onion). To do this, we will use EPIC-KITCHENS [6, 7], the largest dataset in first-person (egocentric) video. While this is primarily a vision dataset, the data includes comprehensive audio associated with each video. As a result of using the audio-stream from a visual dataset, action labels are implicitly provided, meaning an extensive dataset is easily obtained. EPIC-KITCHENS is solely recorded from within the kitchen - interactions within the kitchen provide a rich dataset encompassing a variety of actions and events. Due to the availability of this data we will focus our work on the kitchen, although there is the potential to expand to a broader range of settings. Using this data we will apply state of the art audio classification techniques to our focus on fine-grained events.

Recognition and classification of audio is already quite well established. The DCASE challenge [12] is a popular yearly challenge focusing on the detection and classification of acoustic scenes and events. The DCASE datasets consist primarily of audio from different

scenes, where entrants develop models to predict a scene from its corresponding audio. Additionally, research focused on understanding speech has provided many important ideas that are applicable to audio understanding in general. These lines of work have resulted in many techniques and ideas that work well for their respective domains (e.g. [1, 11, 20, 35]). Whilst recognising fine-grained events exclusively from audio has not been well explored, these related techniques offer an avenue of exploration; they can be applied to our novel setting, allowing for the development of effective fine-grained event recognition, as well as providing an opportunity to develop understanding of these techniques and why they may or may not work in different domains.

1.2 Objectives

The fundamental aim of this project is to develop an effective, audio-based, action recognition model for fine-grained events. By doing this we hope to further understand the impact of audio in the fine-grained events setting, as well as understand how effectively different audio-based techniques transfer to this setting. To fulfil this aim, we have three key objectives. These are:

1. Develop a basic classifier to perform action recognition on the audio-stream of fine-grained events.
2. Apply state of the art audio classification work to the audio-based, fine-grained event setting
3. Understand the performance of audio-based action recognition work in the proposed setting. Identify reasons why it may or may not perform as well in this setting.

1.3 Contributions

This work presents several contributions to the areas of audio understanding and action/event recognition. Primarily, by developing an audio-only model for fine-grained events this work will demonstrate the utility of audio in this setting. Prior work has looked at audio in a multi-modal setting for fine-grained events [17], as well as audio only for acoustic scene classification (e.g. [15, 20]), however, there is no work that we are aware of that considers audio exclusively for fine-grained events. By demonstrating effective approaches for understanding audio for fine-grained events, we also hope that this work may contribute to the use of audio more generally in fine-grained events; multi-

modal approaches that consider audio may be able to utilise this work to maximise the impact of audio in their models.

Additionally, by applying general audio understanding and acoustic scene classification techniques to fine-grained events, we are able to demonstrate the effectiveness of these approaches in multiple domains as well as understand their limitations.

1.4 Outline

Section 2 provides the necessary background for this work, discussing fundamental concepts and relevant literature, as well as describing the datasets used. Section 3 describes our process for developing our models. Section 4 describes the experiments, their results, as well as providing in-depth discussion and analysis. Finally, Section 5 concludes the work and suggests areas for further development.

Chapter 2

Background

2.1 Audio Processing

2.1.1 Spectrograms

Spectograms are a visual representation of a sound wave; they captures temporal, frequency and amplitude information in a single image. Spectrograms are a particularly useful representation because of their ability to represent audio with an image image, allowing conventional image-based techniques to be used on sound. They also provide frequency information that is not directly visible in a standard waveform. Typically, frequency is represented in the vertical axis and the horizontal axis represents time. Amplitude is represented through colour. Spectograms are usually created through performing a Short-time Fourier transform (STFT) on the audio signal. There are several approaches that build on top of the STFT, such as the mel spectogram, which transforms the frequency scale such that sounds that are equidistant from each other also sound equidistant to the human ear. Spectograms may also be converted onto a log-scale as human-audible sounds typically fall into a small range of frequencies

Figure 2.1 shows an example spectogram from the DCASE Urban Acoustic Scene dataset of a recording in a metro station. The recording is primarily indistinguishable background noise with some slightly audible speaking around the 5 second mark, corresponding to a subtle colour (amplitude) change at approximately 3000Hz.

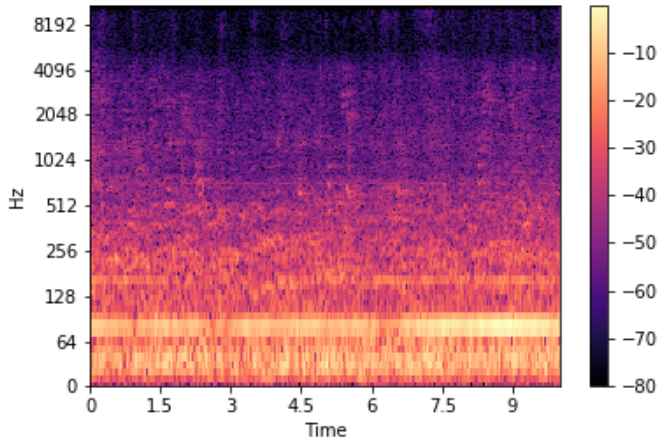


Figure 2.1: *An example spectogram from the DCASE Urban Acoustic Scene dataset. The audio is a 10 second recording of background noise in a metro station*

2.1.1.1 Short-time Fourier Transform Spectrograms

The fundamental component of the spectrogram is the Fourier transform (FT):

$$\int_{t_1}^{t_2} x(t)e^{-2\pi ift}dt \tag{2.1}$$

Here $x(t)$ is a waveform, and $e^{-2\pi ift}$ is Euler's identity, equivalent to a full rotation around a circle at a speed determined by ft. The Fourier transform captures the frequency information of a *stationary* signal - a signal that is constant across time. Whilst useful for any stationary signal, the majority of real-world audio signals are not stationary - they vary across time. This temporal aspect of audio signals is fundamental to understanding an audio signal and, therefore, cannot be ignored.

The STFT is an attempt to resolve this issue of stationarity, allowing informative representations of non-stationary audio to be captured. The basic principle is to split the audio signal into small segments, applying a FT to each segment of the audio and combining the output of each segment to create an image. The STFT assumes that if we split the signal into suitably small segments, the signal within those segments will be stationary. Equation 2.2 gives the definition of the STFT:

$$\int_{t_1}^{t_2} x(t)w(t-\tau)e^{-2\pi ift}dt \tag{2.2}$$

The audio signal is segmented by sliding a small window, $w(t-\tau)$, over the waveform,

$x(t)$. The product of $w(t-\tau)$ and $x(t)$ is then used as the input to the Fourier transform. The window itself is a function that is zero-valued outside of the time range of interest, and is usually symmetric around the point of interest. Commonly a Gaussian or Hann window is used, creating a smooth window edge to avoid noise.

There is an important trade-off to be considered when selecting the size of the window. The standard FT that assumes the entire audio clip is stationary is equivalent to selecting a window of infinite length, giving perfect frequency information, however all time information is lost. By reducing the window size to capture temporal information, the frequency resolution is reduced. There is, therefore, a trade-off between the frequency and time resolutions. A small window will result in a high time resolution, but low frequency resolution, and a large window will result in a low time resolution and high frequency resolution. This trade-off has to be considered so that a useful and informative representation can be found for the specific dataset being used.

When creating a spectrogram representation there are a few key parameters that must be considered; window length and hop size. Window length is the size of the window, w, as seen in Eq.2.2 and hop size specifies the timesteps, t, at which the window is applied to the audio signal. When hop size is equal to window length there is no overlap between each window. A hop size smaller than the window length results in overlapping windows, and a hop length greater than the window length causes the windows to skip over sections of the signal.

2.1.1.2 Mel Spectrograms

The mel spectrogram is a variant upon the STFT that is easier to work with and captures a more human-understandable representation. The mel spectrogram is created by transforming the frequency axis onto the mel scale. The mel scale is a non-linear scale such that sounds perceived by humans to be equally distant are equally distant from each other on the scale. This transformation is performed by splitting the standard (Hz) frequency scale into bins and transforming each of these bins to a corresponding bin in the mel scale. Because of this binning process we now have additional flexibility to determine the dimensions of the frequency axis by selecting the number of bins.

2.2 Deep Learning

Since the introduction of AlexNet in 2012, deep learning has become an extremely popular and effective branch of machine learning. Within computer vision, the state of the art for many important and difficult tasks such as object detection [32], semantic segmen-

tation [33], and image classification [34] use deep learning, primarily with convolutional neural networks [22] (CNN). This trend is also present within audio understanding; the winners of the DCASE scene classification challenge [5, 9, 24, 28] over the last 4 years have all used CNN's.

2.2.1 CNNs

Convolutional neural networks [10, 22] are a type of neural network typically used for computer vision tasks (e.g. image recognition, semantic segmentation). However, CNNs also have some application in other fields, such as audio classification, as we are using them for here. Standard feed-forward neural networks learn weights for each input dimension at each layer. In contrast, CNNs use a fixed kernel of weights that slides over the input (typically an image) and compute the dot-product between the kernel and image as the kernel moves across the image. The output is passed through a non-linearity (e.g. ReLU [2]), and the process is repeated, resulting in a deep network. The network is then optimised through some method of gradient descent, most commonly stochastic gradient descent (SGD) [27], although many other optimisation algorithms are also used (e.g. Adam [18]). In practice, the choice of optimiser is dependent on the application, often requiring experimentation with a few different techniques to determine which is best for the given task.

Instead of inputting a raw image into a CNN, the image is typically split into *channels* (C), usually corresponding to the red, green and blue channels of the input image. As the input propagates through the network, these channels become more abstract and may increase in number. The kernel of weights also consists of channels, one for each of the input channels at the current layer. The total number of parameters in each layer of this network is then given by the number of channels multiplied by the height and width of the kernel: $C * h * w$. Whilst three channels is typical for most images, when working with spectrograms only a single channel is used because the spectrogram is not represented by a typical RGB image, but a single channel image. Because we can so easily represent audio in an image-like representation CNNs are the most obvious choice when it comes to audio understanding.

2.2.2 ResNet

ResNets [14] are a specific CNN-based architecture that utilise the idea of *skip-connections* (Figure 2.2). A skip-connection is an identity mapping from individual layers to later layers in the network, allowing fundamental information from early layers to be easily

propagated throughout the network. The information from earlier layers is incorporated into the output of later layers through a simple summation. The introduction of ResNet was fundamental in training deep convolutional networks, and the architecture is very popular, effective and influenced many subsequent architectures.

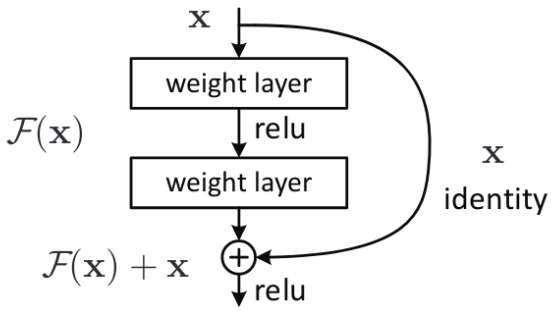


Figure 2.2: *Figure from [14]. The skip-connection and identity mapping introduced by ResNets. This increases the ability of the information in earlier layers to flow into the later layers of the network, allowing deeper networks to better utilise their increased representational power.*

Prior to ResNets, neural networks were limited in how deep they could be; increasing depth generally improved performance, but there was a limit to this, where further increases in depth begin to reduce performance [13]. Consider a pair of networks, one with n layers and the other with $n + k$ layers. The deep network should always be able to perform at least as well as the smaller one by performing a simple identity mapping in the additional k layers. Any additional learnt representation in the k layers would then improve upon the performance of the original network in practice, however, learning such an identity mapping is often difficult, if not impossible for our current gradient-based optimisation techniques. By introducing the skip-connection ResNets make it much easier to achieve this identity mapping for the final k layers of a network and, more generally, make a network easier to optimise.

2.3 Deep Learning for Audio

Outside of computer vision, deep learning is also a very effective tool when working with audio. Traditionally, recurrent neural network (RNN) based architectures, such as the LSTM [16] have been used for audio, particularly for tasks such as speech recognition. RNN's capture temporal information, and maintain an internal state that learns long-term dependencies, making them an obvious choice for many audio tasks. More recently, however, CNNs have been shown to be very effective when working with spectrograms [15], as well as raw waveforms [35].

2.3.1 Wavenet

Spectrogram-based applications of CNNs follow the same paradigm as when working with images - 2D convolutions. The spectrogram can be considered a single channel image, allowing any standard CNN architecture to be used by removing two of the usual three input channels. When working directly with the raw waveform, as in [35], a 1D convolution must be used. [35] is a landmark paper for audio generation, and offers some insight into how deep learning can be used for audio. The paper introduces *causal convolutions*; an approach for capturing temporal dependencies with a CNN. Causal convolutions enforce the temporal ordering of the data by ensuring the prediction at timestep t is only dependent on the data at previous timesteps, $x_1, .., x_{t-1}$ (Figure 2.3)

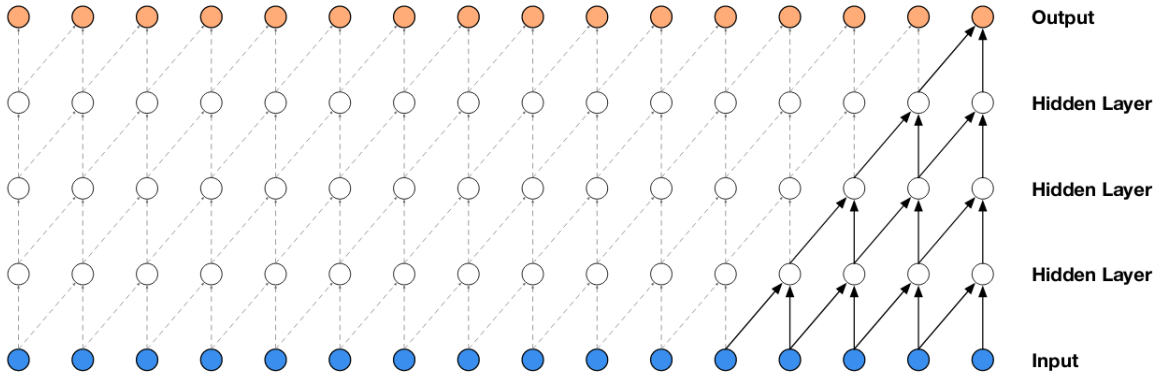


Figure 2.3: *Figure from [35]. The causal convolution architecture. Each node only passes its output forwards in time, and the final layer output becomes the input for the next timestep.*

In order to make each prediction only dependent on previous timesteps, the input nodes in the network are considered sequentially, only feeding the output of a node to later nodes. At training time the entire sequence is already known, so the predictions for the full sequence can be made in parallel. At testing, however, only an initial sequence is known. In this case, the input sequence is fed into the model and the subsequent output is then used as an input for the next timestep. These causal convolutions capture temporal dependencies similarly to how an RNN would, however, because they lack the recurrent connections they are typically quicker, particularly for long sequences.

Whilst quite effective, the authors note that causal convolutions present an issue with a very slowly growing receptive field as a consequence of the reduced number of connections. The receptive field is the region in the network's input that an individual hidden unit is 'looking' at. As the network gets deeper the receptive fields get larger as each

node is considering a larger area of the input. *Dilation* is a way to rapidly increase the receptive field of a network, allowing it to learn higher order abstractions without increasingly computations. Dilation works by padding convolutional kernels with zeros, effectively spreading the kernel out over the image. Figure 2.4 shows how this is applied in the context of causal convolutions.

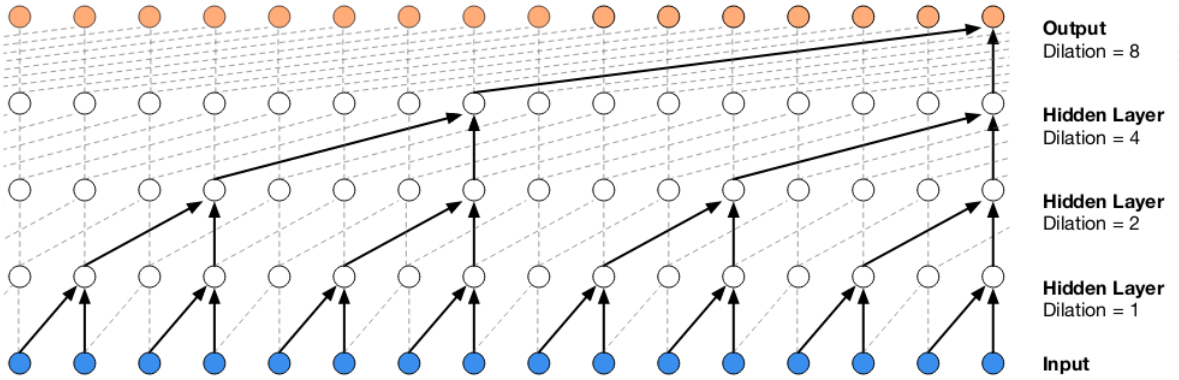


Figure 2.4: *Figure from [35]. The dilated causal convolution.*

2.3.2 Audio Classification

Of particular relevance to this project is audio classification. Looking at the most recent DCASE scene classification competition winners we can get a sense for the general trends in this area of research. [9] use a VGG-net [30] on a spectrogram representation, combined with I-vectors. [24] use a GAN to generate additional training data, and then perform classification by averaging predictions on a spectogram input between a Support Vector Machine (SVM) and feed-forward neural network. [28] consider an ensemble of VGG-inspired networks with spectrogram inputs. Most recently, [5] use GANs to create a data augmentation scheme, and then classify spectrograms using an ensemble of two CNNs. From these works that are a few clear themes: CNNs work well for audio classification, spectrograms are an effective data representation and data augmentation can improve performance. The DCASE acoustic scene classification dataset is quite small (40 hours), explaining why data augmentation is effective in this case.

[15] considers several of the most influential CNN architectures for audio classification (AlexNet, VGG, Inception V3, ResNet50), and compares performance on their proposed YouTube-100M dataset, finding that all the CNN architectures outperform a feed-forward network, and the Inception and ResNet50 architectures achieve the best performance. Not only does this work demonstrate the utility of CNN's for audio classification, it also justifies our choice of the ResNet50 architecture for this project.

[17] performs video classification on the EPIC-KITCHENS dataset, however, they take a multi-modal approach where they also consider audio, as well as optical flow, which they call the Temporal Binding Network (TBN). The TBN receives inputs from three different modalities (audio, RGB and optical flow), and combines these modalities across multiple temporal offsets via a *temporal binding window*. Unlike previous work [23, 36] where each modality is first aggregated temporally, these networks fuse modalities before any temporal aggregation. Their approach had two key takeaways - they demonstrated that fusing modalities through *mid-level* fusion is superior to the previously used late fusion. Specifically relevant to our work, the authors also demonstrated the importance of audio signals for action recognition, showing increases of 5% and 4% through the introduction of audio in previously seen kitchens, and unseen kitchens, respectively. Previous work [37] has suggested that audio is not very informative for action recognition. [17] do find that amongst the three modalities they used (audio, RGB and optical flow), audio is the least informative for action recognition, however, by including audio as a third modality they were able to achieve consistently better results. From these results the authors conclude that audio is a useful modality for action recognition, particularly for fine-grained events as seen in the EPIC dataset. This work sets the context for our project - demonstrating the utility of audio in a multi-modal setting, and also raises the question of how good audio alone is for fine-grained action recognition.

2.3.2.1 Frequency Aware Convolutions

Frequency Aware Convolutions (FAC) [19] is an approach designed to improve the performance of CNN's on spectrograms. A fundamental feature of the convolution is that it is *spatially invariant*; the same visual feature can be recognised regardless of where it appears in the image. Whilst this is advantageous for conventional computer vision tasks (e.g. object recognition), when working with spectrograms this invariance is not desirable. A feature in one location on the spectrogram may sound very different to a similarly appearing feature at a different location if they have different frequency values. Therefore, it is important to break this spatial invariance; FAC do this by including frequency information in an additional channel, allowing the network to distinguish between similar appearing features at different frequency locations, therefore, breaking the spatial invariance of the model. Denoting the pixel values in the spectrogram as (f, t), the additional channel, $V(f, t)$ is defined as

$$V(f, t) = f/F \tag{2.3}$$

where f is the index of each pixel in the frequency dimension, and F the size of the frequency dimension, giving a normalised frequency value corresponding to each pixel in the spectrogram.

The authors of FAC also observe that some more modern deep networks (e.g. ResNet) are not able to translate their increased performance on images to the audio domain [19, 20]. They observe that this issue can be addressed by limiting the size of a model's receptive field, effectively regularising the model and therefore reducing overfitting. In order to reduce the receptive field of a Resnet, the authors consider reducing the filter sizes and reducing the number of layers. They find that there is a sweet spot where limiting the receptive field gives an increase in performance, however, too much reduction is detrimental. In [19], the authors combine this receptive field regularisation with FAC; they note that in addition to breaking the spatial invariance of the CNN, FAC combine well with their receptive field regularisation because by limiting the receptive field, the amount of frequency information available to the model is also reduced. This reduction of frequency information is then resolved by adding an additional frequency channel. The receptive field regularisation performed in [19] is determined by a parameter ρ. ρ specifies the filter sizes, x_k in the first five layers of the network, where k indicates the current layer, and is defined in Equation 2.4. Table 2.1 shows the resultant maximum receptive field for each value of ρ.

$$x_k = \begin{cases} 3 & \text{if } k \leq \rho \\ 1 & \text{if } k > \rho \end{cases} \tag{2.4}$$

ρ value	Max RF	ρ value	Max RF
0	23×23	1	31×31
2	39×39	3	55×55
4	71×71	5	87×87
6	103×103	7	135×135
8	167×167	9	199×199
10	231×231	11	263×263
12	295×295	13	327×327
14	359×359	15	391×391
16	423×423	17	455×455
18	487×487	19	519×519
20	551×551	21	583×583

Table 2.1: *Correspondance between each ρ value and the models maximum receptive field.*

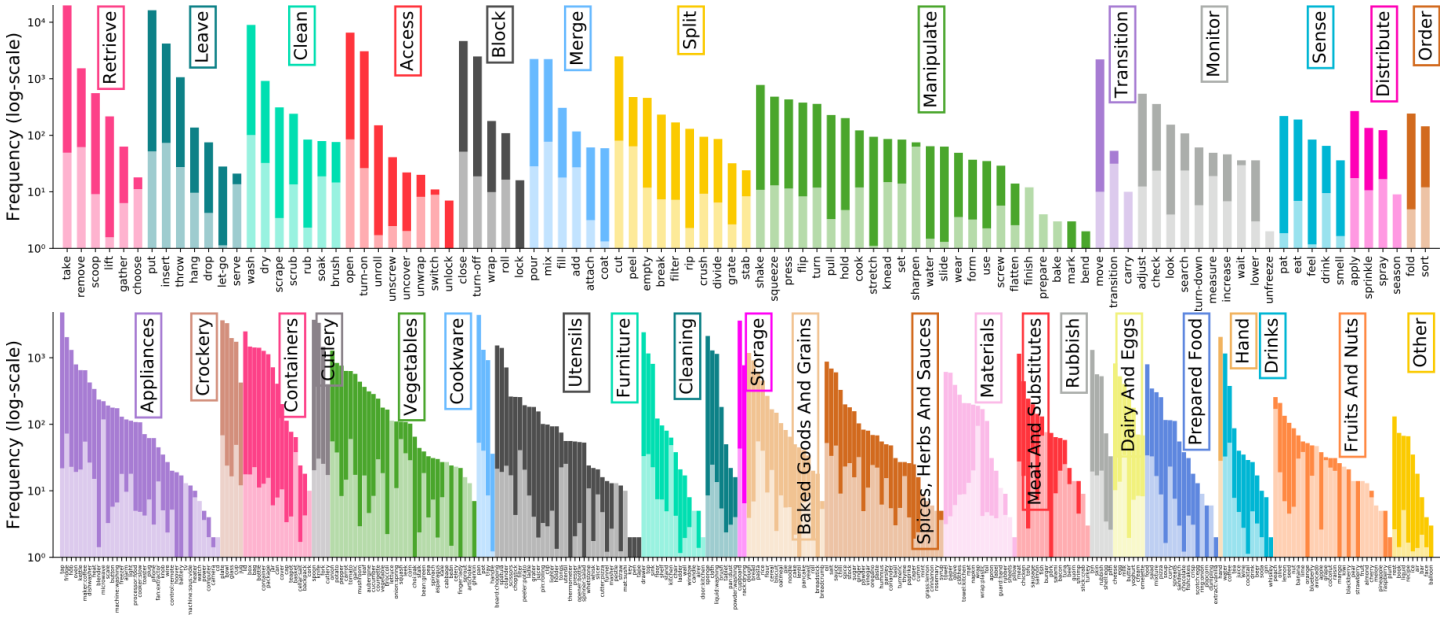


Figure 2.5: Top: *Frequency of action annotations (verbs)* Bottom: *Frequency of object annotations (nouns)*

2.4 Datasets

This work will primarily use the EPIC-KITCHENS-100 dataset [7] due to its focus on fine-grained events. Additionally, the DCASE 2019 [38] dataset is used to test out models and pre-processing and is also used by many related works in the literature.

2.4.1 EPIC-KITCHENS-100

EPIC-KITCHENS-100 consists of 100 hours of visual-audio data from across 45 different kitchens recorded from a head mounted camera, making it the largest dataset in first-person (egocentric) vision. The data is entirely unscripted, limited to within the kitchen and does not include any extra people. EPIC-KITCHENS-100 is a recent extension to the original EPIC-KITCHENS-55 dataset [6]

The dataset features 20M frames, which are labelled to give 90K action segments and 38M object bounding boxes. As we are focusing on audio, the bounding boxes are not relevant to this project. Each label contains a verb and noun pairing with 97 verb classes, and 300 noun classes. Figure 2.5 shows the frequency of both the verb and noun classes across different action and object categories. The training, validation and test sets split the 100 hours of audio into 74.7h, 13.2h, and 12.1h, respectively.

2.4.2 DCASE

A common dataset and challenge in the area of acoustic scene classification is DCASE [12]. For this project we will use the TAU Urban Acoustic Scenes 2019 dataset from Task 1 of the 2019 DCASE challenge [38] . This dataset consists of 40 hours of recordings from multiple acoustic scenes across a range of European cities. Scenes include airports, public squares, and stations. Each data-point is a constant audio signal of about 10 seconds, which we will use for testing baseline models and developing an understanding of pre-processing techniques.

Chapter 3

Methodology

3.1 Data Exploration

A significant proportion of this project has gone into understanding and exploring both the DCASE and EPIC-KITCHENS-100 datasets. In order to gain an understanding and familiarity of spectrograms, we began by experimenting with visualising the DCASE data and training some basic models.

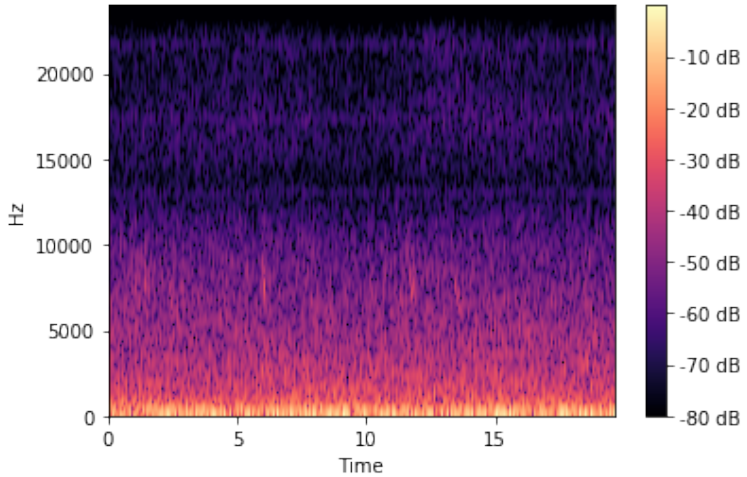


Figure 3.1: *An STFT spectrogram on DCASE with hop and window lengths of 5ms. The audio is from a park in Stockholm*

3.1.1 Exploring DCASE

Initially we considered an STFT with a hop length and window length both of 5ms, resulting in the following spectrogram for the audio of a park in Stockholm (Figure

3.1)

We then compared that to a mel spectrogram of the same recording, hop length and window length, using 128 mel filters (Figure 3.2).

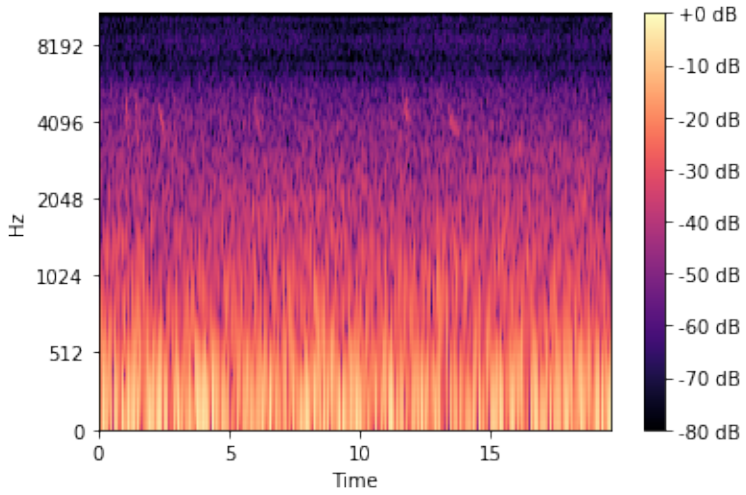


Figure 3.2: *An STFT mel spectrogram on DCASE with hop and window lengths of 5ms. The audio is from a park in Stockholm*

Comparing Figures 3.1 and 3.2, it is clear that the mel spectrogram is more much effective at capturing the lower-frequency bands, as well as extra detail, such as birds squawking at around the 4096 Hz mark. The mel spectrogram still needs tuning, although it is clearly offering a better representation than the standard STFT spectrogram and is, therefore, what we will use going forwards.

The hop size and window length have a significant impact on the spectrogram representation and must be tuned correctly for the dataset. In order to understand their impact we began by adjusting them manually and observing their effect on the representation, and then explored how they impacted performance of a model performing classification on the data. Figure 3.3 shows the same spectrogram as above, but with an increased window length of 20ms. Increasing the window length has decreased the time resolution, and, following the trade-off described in Section 2.1.1, has increased the frequency resolution. Consequently, the small bird noises observable at approximately 4096 Hz are now much more visible.

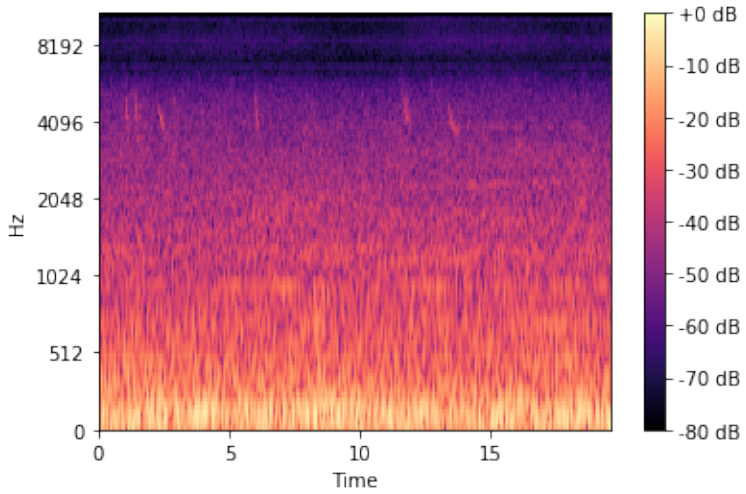


Figure 3.3: *An STFT mel spectrogram on DCASE with a hop length of 5ms and window length of 20ms. The audio is from a park in Stockholm*

Increasing the window size even further continues to increase the frequency resolution, making these higher frequency features easier to distinguish, although comes at a cost of increasingly poorer resolution in the time-dimension. A window length of 20 gives a nice representation that is quite human-readable. Using this window length it was possible for humans to classify many DCASE audio scenes after having seen a few examples from each class. In order to understand the impact of hop size, we then trained a ResNet50 model from on DCASE, adjusting hop size and observing it's impact on performance. The ResNet50 was trained using SGD for 100 epochs, the results can be seen in Table 3.1. From these preliminary results we can observe that a lower hop size gives better performance on the DCASE data. These results are somewhat expected; a lower hop size means there is more overlap between the windows, resulting in more windows, meaning more information is captured.

window length	hop size	top-1 accuracy %
20	5	60.26
20	10	57.08
20	15	53.52
20	20	57.91

Table 3.1: *Top-1 accuracy of a Resnet-50 train on DCASE across a range of hop sizes for a mel spectrogram*

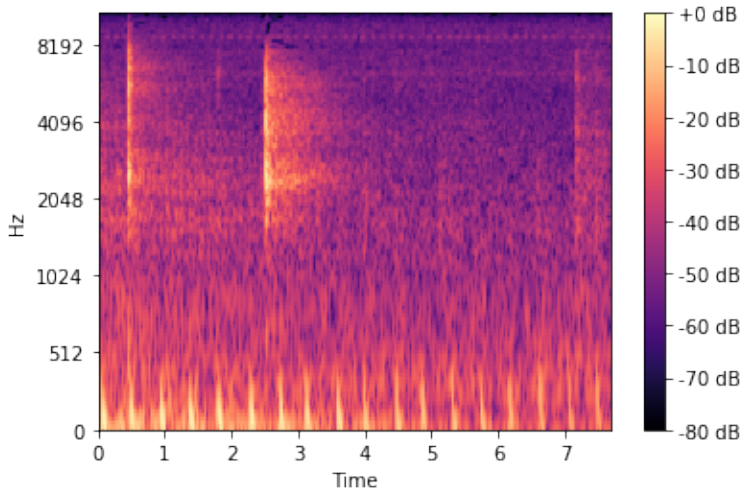


Figure 3.4: *An STFT mel spectrogram on EPIC with a hop length of 5ms and window length of 20ms. The audio is from a plight switch being turned on.*

3.1.2 Exploring EPIC-KITCHENS

After developing an understanding of spectrograms and their application within the DCASE dataset, we shifted our focus to the EPIC-KITCHENS-100 dataset, which is the dataset used for the rest of the project. We began with the same window length and hop size as determined on DCASE, which resulted in quite a nice audio representation (Figure 3.4). After some minor adjustment a window length of 10ms and hop size of 5ms was decided upon, resulting in the spectrogram shown in Figure 3.5. Reducing the window length from 20ms to 10ms gave a slightly sharper resolution in the time axis, making some recordings easier to represent.

The parameters for the spectrograms used throughout the rest of this work are a mel spectrogram with 128 mel filters, a window length of 10ms, and ahop size of 5ms. Using these spectrogram parameters we trained a ResNet50 architecture on the verbs of EPIC-KITCHENS-100 using SGD and a learning rate of 0.001. On the EPIC-KITCHENS-100 validation set this model performed reasonably well, scoring a top-1 accuracy of 38%.

3.2 Developing a Baseline

Having determined our spectrogram parameters we looked to understand the training process on the EPIC KITCHENS data using a ResNet50, with the aim of developing an effective baseline model to compare to the frequency aware convolutions. Previous models on EPIC-KITCHENS [17] use a multi-headed approach where they train a single

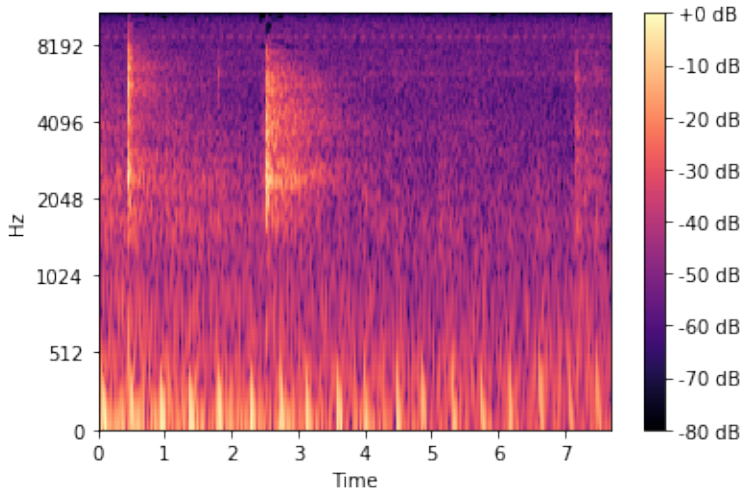


Figure 3.5: *An STFT mel spectrogram on EPIC with a hop length of 5ms and window length of 10ms. The audio is from a light switch being turned on.*

network with two heads; one head predicts verbs and the other predicts nouns. In our case we take a single headed approach and use two models; one for nouns and one for verbs. Most of our experiments have been performed on verbs although the results were also checked to see if the same patterns held for nouns.

3.2.1 Augmentation

We began by exploring the impact of data augmentation on model performance. SpecAugment [25] is a state-of-the-art augmentation approach for spectrograms; the paper takes a similar approach to common image augmentation techniques by applying masks and warping. SpecAugment applies both a time and frequency masking, replacing adjacent columns or rows in the spectrogram with zeros, or alternatively, the mean of the spectrogram. In addition, SpecAugment performs time-warping, where the spectrogram is warped along the horizontal (time) axis. The amount of time-warping is determined by a parameter, W. Given a spectrogram with T timesteps, a random point along the central horizontal line within $(W, T - W)$ is selected and shifted either left or right by an amount w, which is chosen from a uniform distribution from 0 to W. In addition to augmentation, we also use ImageNet [8] pre-training and dropout ($p = 0.5$) [31].

Table 3.2 shows the model performance across a selection of SpecAugment parameters. The results indicate that SpecAugment offers a significant improvement over the original model we trained previously, however, the specifics of the augmentation parameters seems to have little impact on performance. Upon visually inspecting some of the spectrogram representations it became clear that whilst there was little variation between the model

augmentation	time warp	freq. mask	time mask	top-1 accuracy
None	-	-	-	37.98
SpecAug	20	30	30	39.24
SpecAug	20	50	50	39.03
SpecAug	30	30	30	39.05
SpecAug	2	20	20	39.58

Table 3.2: *Results of different augmentation parameters for a Resnet-50 trained on verbs from EPIC-KITCHENS-100*

performance, spectrograms with a large time warp parameter became extremely warped and unrecognisable, prompting an experiment with a time warp value of 2, which gave the best performance. Additionally, larger frequency and time masks produced some spectrograms where the majority of the spectrogram was covered by a mask, so slightly lower values of 20 were selected there as well.

3.2.2 Optimisation

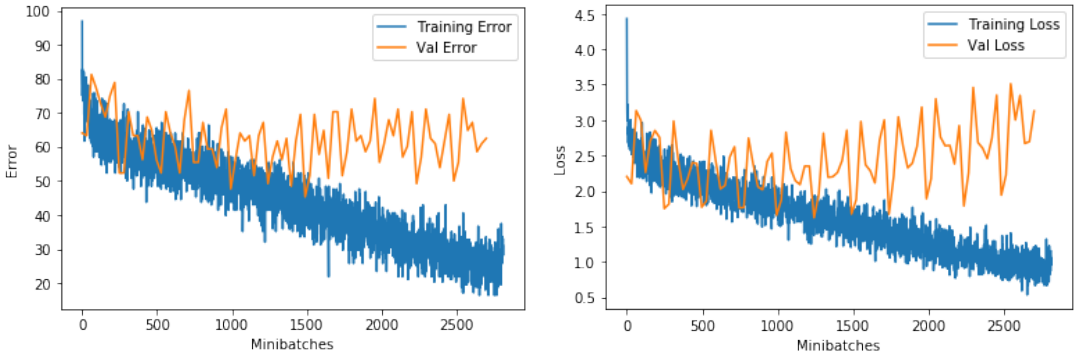


Figure 3.6: *Training and validation error (left) and loss (right) curves using SpecAugment on EPIC-KITCHENS.*

After determining the augmentation setup we began to look at the optimisation process. Figure 3.6 shows the training curves for the best performing augmentation scheme (time warp: 2, freq. mask: 20, time mask: 20), using SGD with a learning rate of 0.0001.

Most importantly, the curves in Figure 3.6 show that the model is learning; both the training and validation loss are decreasing. There is, however, room for improvement with the learning process. Both the error and loss curves show the validation curve initially decreasing with the training curve, and then diverging quite early on. This divergence between the training and validation curves suggest that the model is overfitting. Furthermore, there is a large and sudden change between the initial, steep section of the

loss curve and the rest of the curve, suggesting that the learning rate may be too large. A more preferable curve would be one that is smoother, particularly in the early stages of learning.

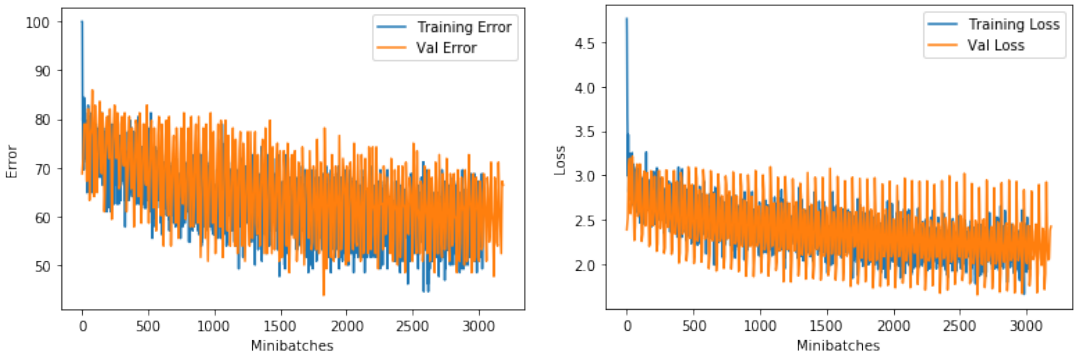


Figure 3.7: *Training and validation error (left) and loss (right) curves with a learning rate of 0.0001 on EPIC-KITCHENS.*

Figure 3.7 shows the same model training with a decreased learning rate of 0.0001. Here the gap between training and validation has decreased, although both curves are now quite flat. This model gave a top-1 validation accuracy of 38.84%. These curves are an improvement upon the previous ones, however, the models clearly stop learning early on in the training process. A way to counter this is through a scheduling algorithm, which adjusts the learning rate throughout the training procedure.

3.2.2.1 Scheduling

There are many different scheduling algorithms that can be used. One of the most popular and simple approaches is to decay the learning rate by some amount at certain specified epochs; this is referred to as a MultiStepLR scheduler[1] in the PyTorch framework [26]. When using the MultiStepLR scheduler one or more epochs are specified at which the learning rate is decayed, as well as how much to decay the learning rate by (γ). We experimenting with this scheduler using $\gamma = 0.1$, which reduces the learning rate by a factor of 10 each time, and tried reducing the learning rate at 20 and 40 epochs, as well as 40 and 60 epochs. Figure 3.8 shows the loss curves for both the 20, 40 run, and the 40, 60 run. From the figure it is clear that the MultiStepLR scheduler has made very little impact on the overall optimisation process. This lack of improvement is confirmed by looking at the validation accuracy for the 20, 40 run and the 40, 60 run, scoring 36.26% and 37.55%, respectively. Whilst it is seems likely that using a learning rate scheduler will improve our model's training, these figures suggest that manually selecting when

[1] https://pytorch.org/docs/stable/optim.html#torch.optim.lr_scheduler.MultiStepLR

to decay the learning rate is a difficult process. Therefore, a scheduling algorithm that dynamically reduces the learning rate may offer improved results.

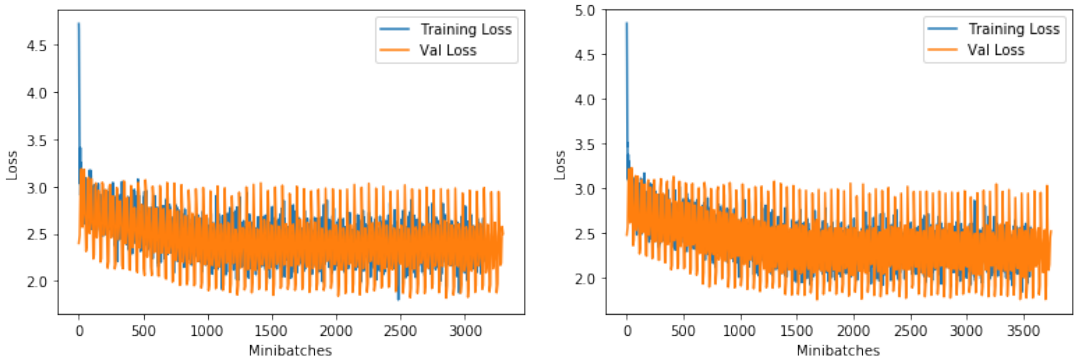


Figure 3.8: *Training and validation loss curves for a MultiStepLR scheduler. Recuding lr at 20, 40 epochs (left) and 40, 60 epochs (right). Both curves have a starting lr of 0.0001 and* $\gamma = 0.1$

One approach to dynamically adjusting the learning rate is the ReduceLROnPlateau scheduler[2]. This scheduler closely tracks the validation loss during training and will reduce the learning rate when the validation loss plateaus according to a specified factor, γ. Figure 3.9[3] shows the training curves using the ReduceLROnPlataeu scheulder. there is already a notable difference between these curves and the ones in Figure 3.8. Primarily, both the loss and errors curves are generally smoother and are learning for longer. Additionally, the gap between training and validation remains small. Comparing Figures 3.9 and 3.6, the impact of a good learning rate scheduler is clear; smoother training and better generalisation. Using the plateau scheduler also resulted in the best performance on the validation set so far, giving a top-1 accuracy of 40.55% on verbs.

[2] https://pytorch.org/docs/stable/optim.html#torch.optim.lr_scheduler.ReduceLROnPlateau

[3] The less prominent validation curves are a result of having to decrease the frequency of model validation in order to speed up training.

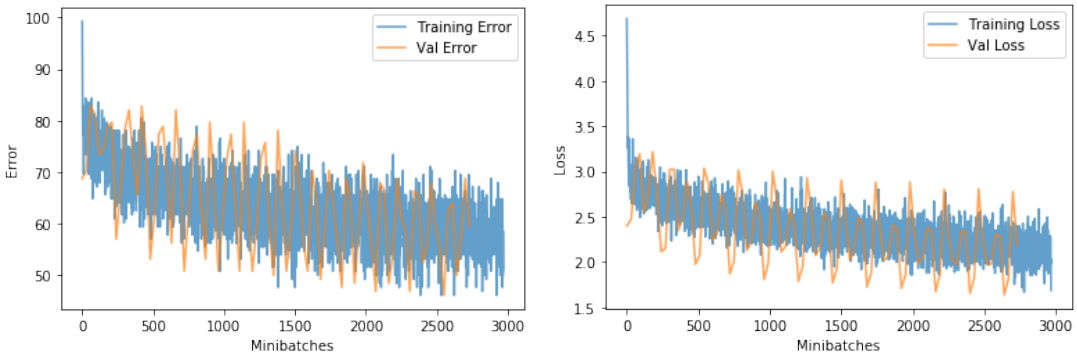


Figure 3.9: *Training and validation loss and curves for a RedueLROnPlateau scheduler on EPIC-KITCHENS. The starting lr is 0.0001 and* $\gamma = 0.1$

3.2.3 Clip lengths

Clip length (s)	Noun/Verb	Top-1 accuracy %
1.279	Verb	40.55
2	**Verb**	**43.71**
5	Verb	39.48
1.279	Noun	18.365
2	**Noun**	**19.27**
5	Noun	17.208

Table 3.3: *Results of different clip lengths for both nouns and verbs. Bold = best*

Neural networks are typically trained in batches. Therefore, it is important that all of the data-points in a batch are of the same size so that they can be represented by a single matrix. For all of the previous exploration the data has been trimmed to 1.279 seconds, as it gives a nice horizontal dimension of 256, and is the same as previous work on the EPIC dataset [17]. Where the full length recording of the action segment is longer than the fixed clip length a random subset is chosen. In cases where the audio clip is shorter than the fixed clip length then audio from the surrounding actions is also included. The addition of information from surrounding actions may give the model more context to classify the recording, however, it may also add uninformative noise from neighbouring action segmments. Additionally, smaller clip lengths may result in cutting out important information from the audio segment. Therefore, it is important to consider which clip length gives the best model performance. Consequently, we experimented with clip lengths of 1.279, 2 and 5 seconds, all of which were using the ResNet50 with SGD, a

learning rate of 0.0001 and the ReduceLROnPlateau scheduler. The results of three different clip lengths for both nouns and verbs can be seen in Table 3.3.

Across the three different clip lengths tried, 2 seconds gives the biggest improvement - over 3% for verbs and 1% for nouns. Using a larger clip length increases the amount of context surrounding the actions, although may also increase the amount of contradictory information from neighbouring actions. By increasing the clip length from 1.279s to 2s the model was able to improve it's performance, most likely because of this additional context; it is less likely that the intended action was cut short, and useful information from neighbouring actions may be incorporated into the spectrogram representation. Increasing the clip length from 2s to 5s, however, was clearly detrimental to performance. The 5s clips were likely too noisy as a result of too much information from surrounding actions, making it hard for the model to discern between the action of interest and its neighbours.

3.2.4 Validation

Finally, in order to most-effectively measure the models validation performance, instead of randomly selecting clips of a specified length as we did during training, we selected multiple clips from the same action segment and averaged the models predictions across these clips. More specifically, when the full action segment was less than the size of the action clip that we were selecting, each action segment was split into 5 clips of equal length (2s). These clips were evenly spaced across the full length of the action segment by calculating the interval, I, between the start of each clip as follows:

$$I = \frac{(t_0 - t_N) - \alpha)}{\beta} \tag{3.1}$$

where α is the fixed clip length (2s in our case), and β is the number of clips that each action segment is split into (5 in our case). t_0 and t_N represent the start and end times of the action segment. Once I is obtained, the validation performance across a single action segment is obtained as follows:

$$\Phi(X_n) = \frac{\sum_0^\beta \Phi(X_n^{0+\beta I, t+\beta I})}{\beta} \tag{3.2}$$

Here, Φ represents the model, making $\Phi(X_n)$ the model's final predictions for the entire action segment and $\Phi(X_n^{0+\beta I, t+\beta I})$ the models predictions for a specific clip within that action segment. $X_n^{0,t}$ would be the first clip from X_n, where t is the clip length. Each clip

is obtained by moving along the full action segment by I. The model makes a prediction for each clip, and the predictions are averaged across all β clips.

3.3 Frequency Aware Convolutions

Our frequency aware convolution implemention is taken directly from the authors code[4]. Their model is built upon a ResNet50 with receptive field regularisation and frequency aware convolutions. The model uses a ρ of 5, giving a maximum receptive field of 87×87, and has been pre-trained on the DCASE dataset.

In order to maintain fair comparisons between FAC and our baseline models the majority of their code was kept the same, with only the data pre-processing and optimisation parameters being changed to match that of the baseline models.

[4] https://github.com/kkoutini/cpjku_dcase19

Chapter 4

Experiments and Results

Model	Overall						Unseen			Tail Classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
TBN(RGB + Flow + Audio)	65.26	47.49	36.08	90.32	73.94	58.04	58.22	38.31	28.36	38.24	26.20	18.89
TSN-Audio	42.63	**22.35**	**14.48**	75.84	**44.60**	**28.23**	35.40	**16.34**	**9.20**	**15.28**	**9.81**	**5.85**
Baseline	**43.72**	20.92	11.50	**78.05**	44.44	23.45	32.50	14.54	8.32	12.19	3.54	2.56
Freq-Aware	40.78	15.96	9.85	77.20	38.17	21.93	**35.82**	11.59	7.35	5.68	1.68	0.71

Table 4.1: *Final results for our baseline, frequency aware model and TSN-Audio, as well as a TBN network using RGB, optical flow and audio, for comparison.* ***bold*** *= best.*

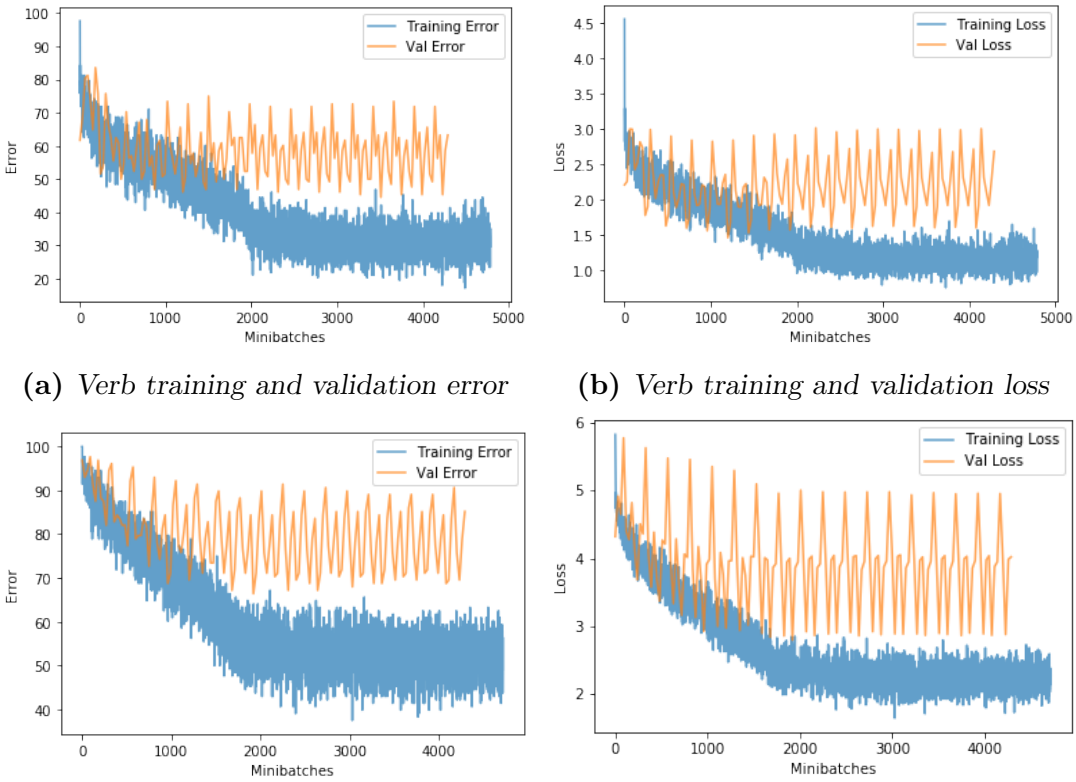
4.0.1 Implementation Details

After determining our data processing and baseline models as described in Section 3, we trained both the baseline and frequency-aware models in order to compare the two approaches. For the baseline model we decided upon the following configuration: A ResNet50 optimised with SGD and a ReduceLROnPlateau scheduler with a base learning rate of 0.0005 and a decaying factor, γ, of 0.1. Each model was trained for 100 epochs. The audio representation is the same for both the baseline and frequency-aware models, for which we used a mel spectrogram with 128 mel filters, a window length of 10ms and a hop size of 5ms. During training a random 2s clip was selected from each action segment where the segment was longer than 2s. When shorter than 2s, a clip of 2s was selected from the mid-point of the segment, resulting in some overlap into neighbouring segments. During validation 5 clips were evenly selected across each action segment and their predictions were averaged.

The frequency aware model was kept the same as in the authors code-base, however, the

optimiser and data pre-processing were swapped out to match the baseline models as described above.

4.0.2 Results



(a) *Verb training and validation error* **(b)** *Verb training and validation loss*

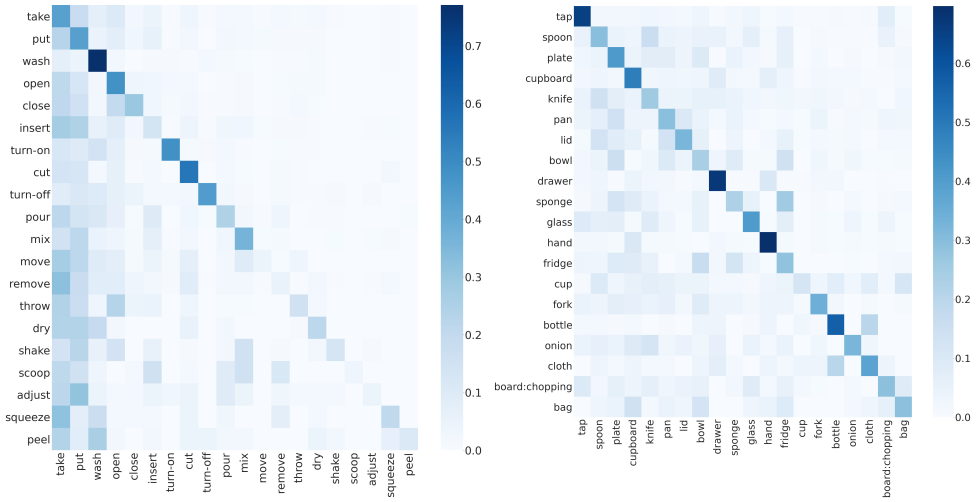
(c) *Noun training and validation error* **(d)** *Noun training and validation loss*

Figure 4.1: *Training and validation error and loss for our final baseline models.*

Figure 4.1 shows the training and validation curves for both loss and error on the baseline models and Figure 4.3 shows the loss and error for the frequency aware model. Table 4.1 shows the training and validation performance on both nouns and verbs for all of our models, as well as a comparison to the multi-modal TBN model, also on the same dataset [7, 17]. Figure 4.2 shows confusion matrices for verbs on nouns on our baseline model and Figure 4.4 shows confusion matrices for the frequency aware model. As well as comparing our baseline and frequency aware models, we also include the performance of a Temporal Segment Network (TSN)[36] trained exclusively on audio. TSN is an activity recognition architecture for video which performs reasonably on our EPIC-KITCHENS dataset.

By examining Table 4.1 we can see that our ResNet50 baseline model achieved the best results on verbs out of the three models, for both top-1 and top-5 accuracy, and had

competitive results across the board. The frequency aware model is performing well, however, it is consistently a few percentage points lower than the other two models, suffering particularly with nouns. The model does, however, score the best out of the three for unseen verbs. We can also compare our results to the TBN models, which use optical flow and RGB video, as well as audio, to classify actions. We can see that whilst the TBN model does significantly outperform all other approaches, the audio-based methods are able to perform reasonably without the additional information from video and optical flow. This result generally supports the findings of [17], demonstrating that audio is a very informative modality, and whilst it is not as effective as multi-modal approaches, action recognition is very much possible from audio alone.



(a) *Verb confusion matrix* **(b)** *Noun confusion matrix*

Figure 4.2: *Confusion matrices for our baseline model on both verbs and nouns*

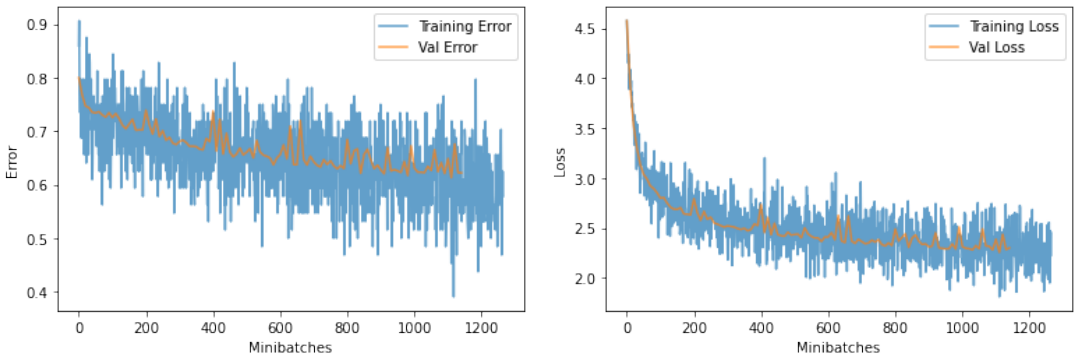
The training curves in Figure 4.1 show a much improved optimisation process when compared to the curves in Section 3. Both the loss and error decrease quite smoothly over time, and there is a notable jump where the scheduler has adjusted the learning rate at around 2000 mini-batches. The gap between training and validation curves is also quite small, suggesting that the model is not excessively overfitting. Additionally, the curves do level off well before training has finished, suggesting that training the models for 100 epochs is not necessary, although this has not harmed validation performance.

Comparing the noun and verb training curves we can see that the noun model appears to be overfitting more than the verb model, and more generally, nouns perform considerably worse than verbs for both top-1 and top-5 accuracy. Whilst nouns generally perform worse than verbs, when visualising classification performance for the top 20 classes with a confusion matrix (Figure 4.2), nouns have a stronger diagonal line, indicating strong performance in the 20 most frequent classes. Because there are significantly more noun

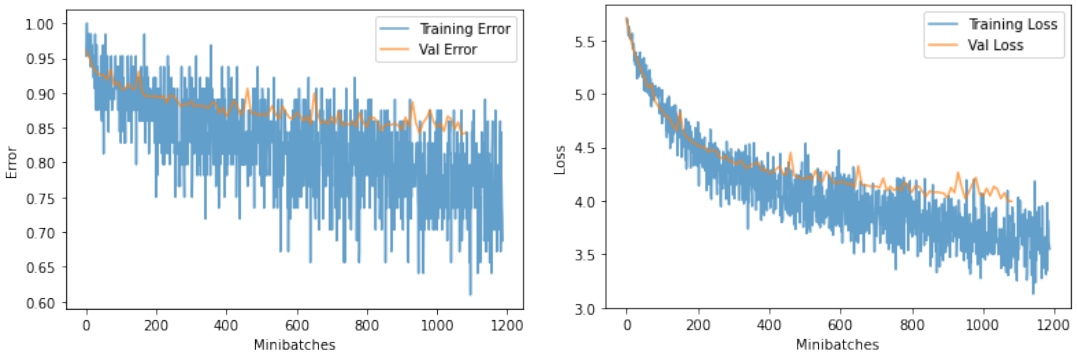
classes (300) than verbs (97), nouns have a large number of classes that do not appear frequently in the data - this is known as a long tail. Because of this long tail it is possible for the nouns to have relatively strong performance on the most frequent classes but still perform poorly overall by failing to accurately classify many of the less frequent classes. Table 4.1 confirms that the noun baseline model performs poorly on the tail classes, with a top-1 accuracy of 3.54%. In comparison, verbs have a top-1 accuracy of 12.19% on the tail classes, which is $\sim$3.5$\times$ better than nouns. When classifying on the entire dataset, however, verbs are only $\sim$2$\times$ better than nouns, confirming that the noun models struggle disproportionally on tail classes. Intuitively, the results for verbs and nouns make sense. Verbs are generally easier to distinguish from audio alone than nouns; many verbs are quite distinct, such as chop, pour, open, close, and will make similar sounds for a wide range of objects (nouns). Nouns, however, can be more ambiguous. Many objects can be acted upon in a wide variety of ways, all of which may sound extremely different. Despite this, there will be some nouns that are substantially easier to classify. These are nouns that have very few uses, such as "tap", where almost all interaction will be turning the tap on or off. In either case, the audio clip will begin or end with the sound of running water and will follow or be followed by the turning of a tap. From examining Figure 4.2 we can see that many of the top 20 classes, which nouns are performing quite well on, have reasonably distinct use cases.

Figure 4.3 shows the training curves for both verbs and nouns of the frequency aware model[1]. The frequency aware training curves stand out in two key ways. Firstly, the training is very smooth, and there is no extreme change in the slope of the curve as the model learns. Secondly, the gap between training and validation is very small; this is particularly notable in the case of verbs, where the validation curves sits almost exactly in the middle of the training curve. One of the key features of the frequency aware model is the receptive field regularisation, which reduces the size of the models receptive field by limiting the filter size of the model. The receptive field regularisation is motivated by the observation that many modern networks typically overfit audio data. If we compare Figure 4.3 to Figure 4.1, the effect of the receptive field regularisation is clear; the gap between training and validation is much larger in our baseline model than in the frequency aware model. However, despite reducing the gap between training and validation, the frequency aware model does not perform as well as our baseline model. The results in Table 4.1 show that whilst the frequency aware model has competitive overall verb

[1]The validation curves appear less noisy than in previous figures because only the average validation performance across the whole epoch is reported, not the performance of each individual batch, as can be seen in previous Figures. This is a consequence of the code provided by the authors of [19]



(a) *Verb training and validation error* **(b)** *Verb training and validation loss*



(c) *Noun training and validation error* **(d)** *Noun training and validation loss*

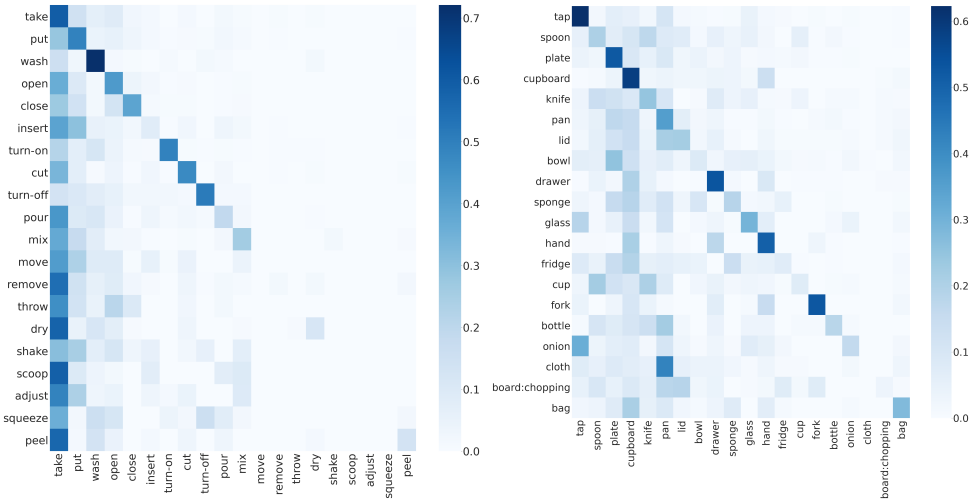
Figure 4.3: *Training and validation error and loss for our final baseline models*

performance, the model particularly struggles with nouns, and tail classes. However, the model does achieve the best performance on unseen verbs; these are participants that do not appear in the training set. The good performance on unseen data may be a result of the receptive field regularisation, reducing the extent to which the model is overfitting to the participants seen when training. Whilst the frequency aware model appears to have effective regularisation, this may be coming at a cost of reducing the amount of information available to the model and impacting the model's performance on nouns and tail classes. The receptive field of the model may capture additional context, which may be necessary in order to effectively classify nouns. By reducing the receptive field it is possible that the model's performance on nouns has been degraded.

The impact of the additional frequency information provided by the frequency channel is unclear, however, given that the frequency aware model outperforms all other models for unseen verbs, it is possible that the additional channel is informative in this setting. By including the frequency channel the model may find it easier to classify verbs which have not previously been seen, however, if this is the case it does not carry over to nouns. The authors of [19] argue that the additional frequency information provided by the frequency channel is necessary to counter for the information lost by reducing the

receptive field. Whilst the frequency aware model is performing well in some cases, the argument that receptive field regularisation and an additional frequency channel pair well together is not supported by our findings, suggesting that the findings of [19] do not carry over to the fine-grained, egocentric setting.

Given that the frequency aware model has very smooth learning curves that do not overfit, it is likely that the model has the potential to improve its performance on the data. The hyper-parameters used for the frequency aware model are taken directly from the baseline model, for which they were carefully tuned (Section 3). It is likely that undergoing a similar process for the frequency aware model may result in better results. In particular, the authors of [19] showed that the receptive field regularisation has the potential to both overfit and underfit the data if too little or too much regularisation is occurring. Our model uses a ρ of 5, which is inline with what the authors suggested works well for their ResNet model on DCASE, however, it is possible that a different value may perform better for our use-case.



(a) *Frequency aware verb confusion matrix*

(b) *Frequency aware noun confusion matrix*

Figure 4.4: *Confusion matrices for our frequency aware model on both verbs and nouns*

Figure 4.4 shows confusion matrices for the frequency aware model on both verbs and nouns. The confusion matrices generally support the results in Table 4.1, showing that the frequency aware model performs worse than than baseline model. The verb confusion matrix shows that the frequency aware model overwhelmingly predicts the majority class, "take", resulting in the poor confusion matrix. The noun confusion matrix still has a strong diagonal, however, this is much noisier than in the baseline model. As seen in the confusion matrices for the baseline model (Figure 4.2), nouns appear to have a stronger

confusion matrix than verbs, despite having worse overall performance. This is again explained by the noun class having a long tail, which the model performs poorly on, despite it's apparently strong performance on the more frequent classes.

Chapter 5

Conclusions

In this work we conduct an extensive analysis of fine-grained, audio-based action recognition using ResNets, frequency aware convolutions and receptive field regularisation. In particular, we explore the impacts of optimisation, data representation and augmentation on a ResNet50 for audio-based action recognition, developing an effective model that gives competitive results. We also consider an effective strategy for scene classification - frequency aware convolutions paired with receptive field regularisation - and apply them to the context of fine-grained audio based action recognition. We demonstrate that whilst this approach is effective it can be outperformed by a well-tuned ResNet50 architecture. We also show that whilst audio only approaches are not as effective as a multi-modal approach such as [17], audio only models are quite capable of performing reasonably at fine-grained action recognition, and with further advancements in the area may become more competitive as a stand-alone approach.

Using a ResNet50 model we explored a few areas that allow it to perform comparatively, and better than, more complex approaches. Primarily, effectively configuring the optimisation process is shown to be very important to develop an effective model. In particular, the pairing of Stochastic Gradient Descent with a ReduceLROnPlateau scheduler proved to be very effective for our use-case. Establishing an informative data representation was also very important for developing our model. The addition of SpecAugment gave considerable improvements, and after carefully selecting the parameters we obtained an increase in accuracy of $\sim 2\%$ compared to no data augmentation. Significant performance gains were also achieved by increasing the length of action clips from 1.28 seconds to 2 seconds. By increasing the length of the audio, additional information was available to the model. However, it appears that there is a limit on how much the lengths can be increased; as the length increases the amount of useful information gained becomes

inconsequential when compared to the additional noise introduced by the neighbouring action segments. Combining these approaches we were able to create an effective model using only a standard ResNet50 which also became a good baseline to compare our frequency-aware model to. Our results were in line with previous work [17], generally finding that nouns are harder to classify than verbs. We analysed the results with confusion matrices and found that whilst nouns performed worse overall, they performed better on the most frequent classes, highlighting the impact of the long tail seen in the EPIC-KITCHENS dataset which is particularly prevalent for nouns.

We also trained a ResNet based architecture using frequency aware convolutions and receptive field regularisation. This architecture is specifically designed to combat issues of overfitting often seen in deep architectures when applied to audio [20]. We demonstrate that the frequency aware, receptive field regularised model does transfer well to our fine-grained action recognition setting, and achieves good results, particularly with verbs. The model, however, does not perform better than our baseline. It is likely that this is because the learning parameters have not been tuned for the frequency aware model, and were determined using the ResNet50 baseline. We examine the optimisation process and results, and observe that whilst the frequency aware model does not outperform our baseline, the training curves are very smooth and appear to generalise well, showing only a small gap between the training and validation curves, suggesting that the receptive field regularisation is effectively reducing overfitting.

In conclusion, in this work we developed two effective approaches for performing audio-based, fine-grained action recognition on the EPIC-KITCHENS dataset, one using a standard convolutional neural network, and the other applying state of the art audio classification techniques to our fine-grained setting. By doing this we demonstrated the utility of audio alone as an important and informative modality for performing action recognition. We also showed that a well-tuned ResNet50 architecture is able to outperform the more sophisticated, state of the art audio classification techniques. This suggests that whilst these action recognition approaches may transfer well to the setting of fine-grained events, they are not necessarily as competitive as they are on their target task. By demonstrating the efficacy of audio-based action recognition we also highlight that whilst not as effective as video, or multi-modal approaches it is very possible to classify actions from audio alone - this has important privacy and security implications that should be considered.

5.1 Further Work

This work considers two different approaches to fine-grained, audio-based action recognition. These are a well-tuned ResNet50 architecture, and a ResNet based model using frequency aware convolutions and receptive field regularisation. Whilst both of these approaches were shown to be effective, the simpler ResNet50 model outperformed the more sophisticated frequency aware model. There is, therefore, lots of room to explore how and why the frequency aware model, whilst effective, has not been able to outperform the simpler baseline. Further work, therefore, should initially focus on understanding and improving the performance of the frequency aware model. There are two clear directions here; firstly, by carefully tuning the model as we did with the ResNet50 baseline in Section 3, there is a lot of potential for improved results. It may also be valuable to experiment with the receptive field regularisation parameter, ρ. This parameter was kept the same as in the original paper, however, the authors were using a different dataset, so it may be valuable to adjust it to our use-case. Secondly, it may be informative to decouple the two fundamental components of the frequency aware model; frequency aware convolutions and receptive field regularisation. By decoupling the two components we can gain an understanding of the impacts and contributions of each. The original codebase provided by the authors[1] contains a receptive field regularisation model with and without frequency aware convolutions. By training the model without the frequency aware convolutions we can gain an understanding of their contribution. It would also be informative to implement frequency aware convolutions and receptive field regularisation on our ResNet50 baseline model, allowing clear comparisons to be made between the ResNet50 baseline, a ResNet50 with receptive field regularisation, a ResNet50 with frequency aware convolutions, and a ResNet50 with receptive field regularisation and frequency aware convolutions combined.

It is also of interest to understand how frequency aware convolutions and receptive field regularisation may apply to the wider domain of multi-modal action recognition, of which audio is a core component. Work such as [17] demonstrates the importance of audio for multi-modal action recognition on the EPIC-KITCHENS dataset. If frequency aware convolutions and receptive field regularisation are able to improve the performance in the context of audio alone, then it seems probable that their inclusion in a multi-modal context would be valuable. In order to understand the impact of the frequency aware model in this case the further work proposed above should be carried out first, as the benefits of frequency aware convolutions and receptive field regularisation must

[1] https://github.com/kkoutini/cpjku_dcase19

be more clearly demonstrated on audio alone, before being considered in a multi-modal context.

There is also scope to consider the application of alternative audio classification techniques to our setting. One approach would be to work directly with a raw wave-form, using standard 1D convolutions, comparing these to causal convolutions from the WaveNet architecture [35]. Unlike receptive field regularisation, WaveNet tries to increase the receptive field size through the use of dilation. Comparing the efficacy of a WaveNet like approach to receptive field regularisation may be informative to understand the sorts of architectures that work well in our domain, and whether the hypothesis of [20], that receptive field regularisation resolves the overfitting seen in many audio models holds true. An additional approach that could be considered is to *time-slice* the spectrograms [1, 11]; this is where the spectrogram is split into columns, where each column represents an individual data point, enforcing the temporal nature of the data. Time slicing and causal convolutions could then also be combined with each other, as well as the approaches used in our work, or all three approaches could be considered together. Both causal convolutions and time-slicing spectrograms attempt to strengthen the temporal aspect of the audio, which is not seen with our approaches, and may be beneficial to create an effect audio-based, fine-grained action recognition model.

Finally, the security and privacy implications of audio-only action recognition should be considered. The possibility for many smart-home devices to determine our actions is very real. The implications of this should be carefully considered before any such technology is deployed on these devices.

References

[1] T. Afouras, J. S. Chung, and A. Zisserman. The conversation: Deep audio-visual speech enhancement. *arXiv preprint arXiv:1804.04121*, 2018. 3, 37

[2] A. F. Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 8

[3] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017. 1

[4] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems*, pages 892–900, 2016. 1

[5] H. Chen, Z. Liu, Z. Liu, P. Zhang, and Y. Yan. Integrating the data augmentation scheme with various classifiers for acoustic scene modeling. Technical report, DCASE2019 Challenge, June 2019. 8, 11

[6] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. i, 2, 14

[7] D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020. i, 2, 14, 28

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 20

[9] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer. CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks. Technical report, DCASE2016 Challenge, September 2016. 8, 11

[10] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 1, 8

[11] A. Gabbay, A. Shamir, and S. Peleg. Visual speech enhancement. In *Interspeech*, pages 1170–1174. ISCA, 2018. 3, 37

[12] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M. D. Plumbley. Detection and classification of acoustic scenes and events: An ieee aasp challenge. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE, 2013. 2, 15

[13] K. He and J. Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015. 9

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. i, iii, 8, 9

[15] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017. 3, 9, 11

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 9

[17] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3, 12, 19, 24, 28, 29, 34, 35, 36

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8

[19] K. Koutini, H. Eghbal-zadeh, and G. Widmer. Receptive-field-regularized cnn variants for acoustic scene classification. In *Preprint*. i, 12, 13, 30, 31, 32

[20] K. Koutini, H. Eghbal-zadeh, M. Dorfer, and G. Widmer. The Receptive Field as a Regularizer in Deep Convolutional Neural Networks for Acoustic Scene Classification. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, A Coruña, Spain, 2019. i, 3, 13, 35, 37

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1, 8

[23] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 12

[24] S. Mun, S. Park, D. Han, and H. Ko. Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane. Technical report, DCASE2017 Challenge, September 2017. 8, 11

[25] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019. 20

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019. 22

[27] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. 8

[28] Y. Sakashita and M. Aono. Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions. Technical report, DCASE2018 Challenge, September 2018. 8, 11

[29] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 2

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 11

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 20

[32] M. Tan, R. Pang, and Q. V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020. 7

[33] A. Tao, K. Sapra, and B. Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020. 8

[34] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou. Fixing the train-test resolution discrepancy: Fixefficientnet. *arXiv preprint arXiv:2003.08237*, 2020. 8

[35] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016. iii, 3, 9, 10, 11, 37

[36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 12, 28

[37] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue. Multi-stream multi-class fusion of deep networks for video classification. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 791–800, 2016. 12

[38] H. Zhu, C. Ren, J. Wang, S. Li, L. Wang, and L. Yang. Dcase 2019 challenge task1 technical report. *Tech. Rep., DCASE2019 Challenge, Tech. Rep*, 2019. 14, 15