Project Report: Fast and Memory-Efficient Coarse-Grained Audio Classification using Deep Neural Network

Md Shamim Hussain
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th St.
Troy, NY, USA
hussam4@rpi.edu

Nafis Neehal
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th St.
Troy, NY, USA
neehan@cs.rpi.edu

Abstract

Deep neural networks are notorious for their high computational cost and memory requirements. Not only that, often they need specialized hardware for efficient execution. However, for simple tasks such as a coarse-grained classification of sound into superclasses such as speech, music, and noise, one may not be able to spare a lot of computational resources. In this work, we devise a new convolutional network - SwishNetV2 which is fast and memory-efficient, but gets close to big convolutional networks such as VGG16 in terms of classification performance. This architecture improves upon the previous SwishNet architecture and performs well on the CPU with limited memory requirements, similar to its predecessor. We verified the performance of this model on the diverse Audioset dataset, with manually verified gold standard labels. We also devised a method for including data with unchecked noisy labels from AudioSet which improves classification performance. Our annotated dataset is available publicly at `https://www.kaggle.com/snirjhar/audioset-speech-music-noise`. The code for our implimentation can be found at `https://github.com/shamim-hussain/audioset_coarse_grained_classification`.

1 Introduction

Music, speech and noise classification and segmentation is an important task because these three types of signals are inherently different in nature [1] and require different types of processing and/or coding schemes. For example, a good compression scheme for speech may not be good for compressing music. Also, the signal processing steps used for these two types of media are likely to be very different. So, correctly identifying the type of media before further processing should be considered a very important task.

With the advent of widespread use of the internet, it is now possible to build large media databases from user-contributed data. However, labels for the collected data are rarely available. Manual media labeling can be both expensive and time-consuming. So, a reliable method is required to automate the indexing of large media databases. It is intuitive to first classify/ segment the media into broad categories such as speech, music and noise which may be followed by further fine-grained classification/segmentation procedure (for example, into different musical genres).

The above discussion suggests that it is desirable to bring the classification/segmentation of media into speech, music and noise under a single framework. In recent years deep learning [2] algorithms have achieved unprecedented success in numerous classification problems, without any need for

Preprint. Under review.

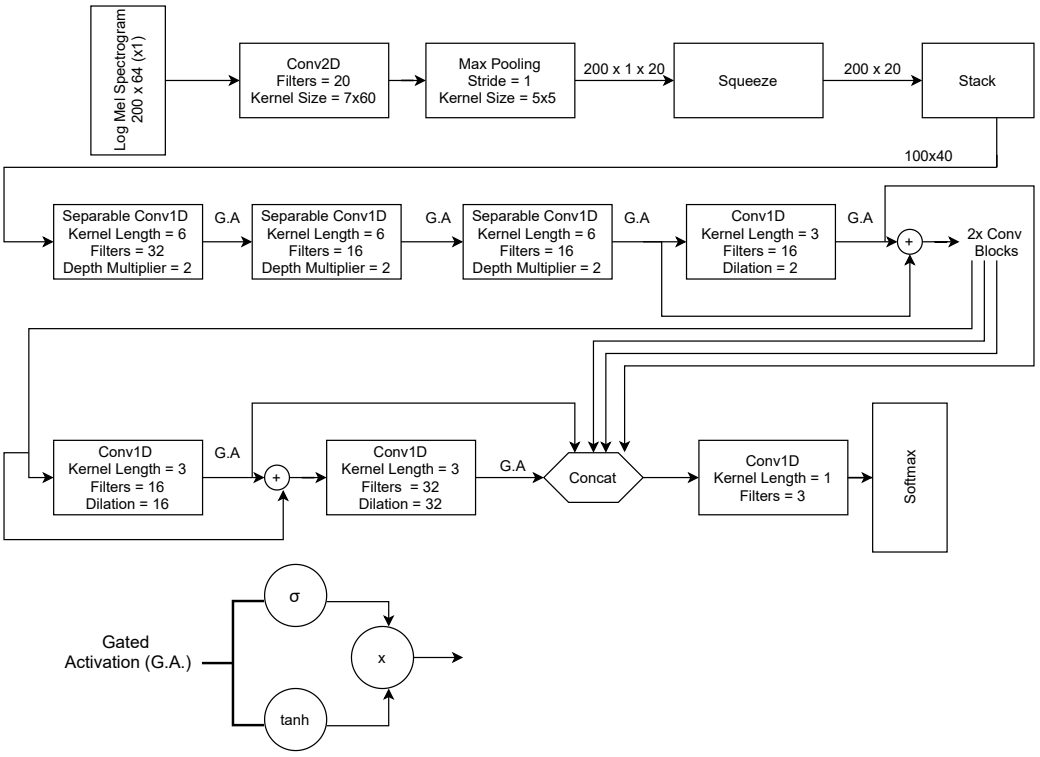

Figure 1: Network Architecture of SwishNetV2

careful feature selection. Especially, deep convolutional neural networks have set new frontiers in many fields such as image and audio classification and segmentation. However, deep neural networks are generally known to be more computationally expensive and slower than other more conventional models. These models often need large-scale parallelization in GPUs for faster implementation. Since classification/segmentation is often set as a pre-processing step in an audio processing pipeline, it needs to be fast enough to prevent the delay in further processing. Also, the computational resources are limited in many scenarios such as in mobile devices and embedded systems where it is unreasonable to waste too much resource in the preprocessing step. This is where we put forth our contribution.

Recently, it has been shown that for coarse grained audio classification tasks a fast and lean 1D CNN applied on derived features can show performance close to that of a 2D CNN - the SwishNet architecture [3]. Although, this architecture has shown good performance on a clean and less diverse dataset such as the MUSAN [4] dataset, it's performance on a more difficult and diversified realistic setting has not been verified. In this work, we further explored the use of lightweight CNN architecutres for audio classification task but in a more diverse and realistic setting - on the AudioSet dataset [5], which is a dataset of 2.1 million annotated audio clips derived from youtube videos. The AudioSet ontology contains 527 different classes, but it does not directly annotate clips into suprerclasses such as speech, music and noise. Instead, it labels each clip with a set of subclasses. We defined these superclasses in the ontology which gives us plausible superclass labels for the clips. However, due to mistakes/missing annotations the actual superclass maybe wrong/ambiguous. So we, manually annotated a set of gold standard labels for around 12000 clips. We devised a new network architecture SwishNetV2 which is an improvement on the SwishNet architecture and directly applies on log mel spectrogram. We verified the performance of this model on the AudioSet dataset with manually verified gold standard labels. We also developed a method for including data with unchecked noisy labels from AudioSet which improves classification performance.

2 Related Works

In general, audio content analysis in video parsing can be considered in two directions [6, 7]. One is to discriminate audio streams into different classes such as speech, music, noise etc., the other is to classify audio streams into segments of different speakers. In this paper, our research work in the first direction will be presented.

There have been many studies on audio content analysis, using different features and different methods [8] - Pfeiffer et al [9], presented a theoretic framework and application of automatic audio content analysis using some perceptual features. Saunders [10], presented a speech/music classifier based on simple features such as zero crossing rate and short time energy for radio broadcast. When a window size of 2.4s was used, the reported accuracy rate would be 98%. Scheirer et al [11] introduced many more features into audio classification and performed experiments with different classification models including GMM (Gaussian Mixture Model), BP-ANN (Back Propagation Artificial Neural Network) and KNN (K-Nearest Neighbor). When using a window of the same size (2.4s), the reported error rate would be 1.4%. However, it is found that such simple feature-based methods cannot work well when a smaller window is used or more audio classes such as environment sounds are taken into consideration.

Many other works have been done to enhance audio classification algorithms. In [12], audio recordings are classified into speech, silence, laughter and non-speech sounds, to segment discussion recordings in meetings. The accuracy of the segmentation result using his method varies considerably for different types of recording. In the work by Zhang and Kuo [13], pitch tracking methods are introduced to discriminate audio recordings into more classes, such as songs, speeches over music, with a heuristic-based model. Accuracy of above 90% is reported. Srinivasan et al [5], try to detect and classify audio that consists of mixed classes, such as combinations of speech and music together with background sound. The accuracy of classification is over 80%.

3 Methodology

3.1 Data Collection

The Audioset [5] ontology was used to identify fine-grained sound event classes (source). We combined the fine-grained event classes into 3 major superclasses - speech, music and noise. We define each superclass uniquely by saying that a particular audio segment must not contain events from other superclasses, so that there is no ambiguity in classification.

We downloaded the annotations of the YouTube video segments. Each segment is assigned multiple sound event tags. We defined plausible superclasses, based on these tags. If a clip contains only the subclasses of a given superclass (speech/music/noise) it is said to belong to that plausible superclass. If the superclass is ambiguous, we ignore that clip. But due to mistakes in annotations, missing annotations and human errors, the actual superclass may still be ambiguous or even wrong. So, the labels assigned at this point are noisy.

Next, we downloaded YouTube clips in uncompressed wav format which was later compressed by the free lossless audio codec (FLAC). Clips were downloaded from the evaluation set, the balanced training set and the unbalanced training set. We downloaded about 83,600, 10s clips from YouTube.

To generate ground truth (gold standard) annotations for the clips we manually checked each clip and verified their plausible superclass. If we found that a clip's superclass was ambiguous/wrong, it was tagged as "bad", otherwise it was tagged as "good". We performed these manual annotations until we reached 4,000 "good" clips for each superclass (speech, music, and noise). Thus, the gold standard ("good") dataset contains 12,000 annotated unambiguous ground truth labeled clips. Also, the ambiguous ("bad") labels are also kept for possible future use in automated label clean up. Our annotated dataset, along with unchecked clips is available publicly at `https://www.kaggle.com/snirjhar/audioset-speech-music-noise`.

3.2 Feature Extraction

We extracted log mel spectrogram features and energy for each clip. Before that, silence portions of the audio clips were removed, since we do not consider silence in our classification task. Our derived

features can be found at `https://www.kaggle.com/snirjhar/audioset-derived-features`. The MFCC features used in [3] can be easily derived from these features.

3.3 Network Architecture

While 1D convolutional networks, like WaveNet [14] can operate on raw audio, we trained our network on frame-wise extracted log mel spectrogram because processing raw audio input is infeasible in terms of computational cost and memory requirement for our problem. Note that SwishNet [3] operates on MFCC features, which is more specialized but we found that log mel spectrogram features are more informative and thus results in better classification performance.

While SwishNet is a 1D convolutional network which operates on frame-wise MFCC features, our model has a single 2D convolutional base followed by 1D convolutions. The rationale behind this is that, the 2D convolution mimics a feature extraction stage, similar to MFCC but at the same time it is trainable. Max Pooling allows for slight invariance in both time and frequency dimensions. Similar to SwishNet, 1D convolutions are carried out along the temporal dimension only. The detailed network architecture of SwishNetV2 is shown in Fig. 1.

After the 2D convolutional base we stacked two halves of the signal which effectively halves the time dimension and saves computational time. Similar to the original SwishNet architecture which follows the WaveNet [14] architecture, we used multiple layers of causal convolutions to gradually increase the receptive field and gated activation functions [15] containing sigmoid and tanh functions instead of widely used ReLU activations. The gated activations allow the network to select which information to pass from one time step to the next, just like a gated recurrent network. It also conveniently cuts down the number of feature maps passing from one layer to the next to half.

At lower layers we used longer filters. Since, longer dense filters are computationally and parametrically costly, here we used separable 1D convolutions [16] instead of dense convolution. These longer filters are followed by a series of dialated convolutions which exponentially increases the receptive field of the network.

3.4 A 2D Convolutional Neural Network for Comparison - VGG16

The most widely used deep learning technique for audio classification is to apply a 2D convolutional network on the spectrogram or other derived features [17, 18, 19]. However, to the best of our knowledge, no other deep learning scheme has specifically addressed classification/segmentation into all of the three classes - Music, Speech and noise under a single unified framework. So, for comparison, we sought an established 2D architecture that achieves highest performance for the task at hand.

For 2D convolutional networks, we treated the log mel spectrum as a 2D signal and convolutions were done along both frequency and time axes. Among the CNN architectures, we considered, VGG16 [20] performed the best for the task at hand especially its initialized with ImageNet weights. It converges much faster and achieves higher accuracy when initialized with ImageNet weights rather than with random weights. So, this is a case of transfer learning [21] from the ImageNet dataset to the AudioSet dataset. To apply the pre-trained networks on log MFB spectrum, we copied the input along three input channels (Red, Green and Blue) which is equivalent to using a greyscale image. We only kept the convolutional parts of the networks and applied Global Average Pooling to the outputs. Then, we applied two dense layers ending in a softmax output.

4 Results and Discussions

4.1 Description of the Dataset

The AudioSet dataset [5] is compiled from annotated youtube videos, which must be collected from original youtube IDs. Often, if the content is not available on youtube, that particular clip must be ignored. It consists of approximately 5.8 thousand hours of audio in 527 categories.

We downloaded around 83.6k audio clips. We encoded all the audio files into 22 kHz FLAC format. Our gold standard dataset contains 12,000 annotated ground truth labeled clips. There were also 66.5k unchecked clips.

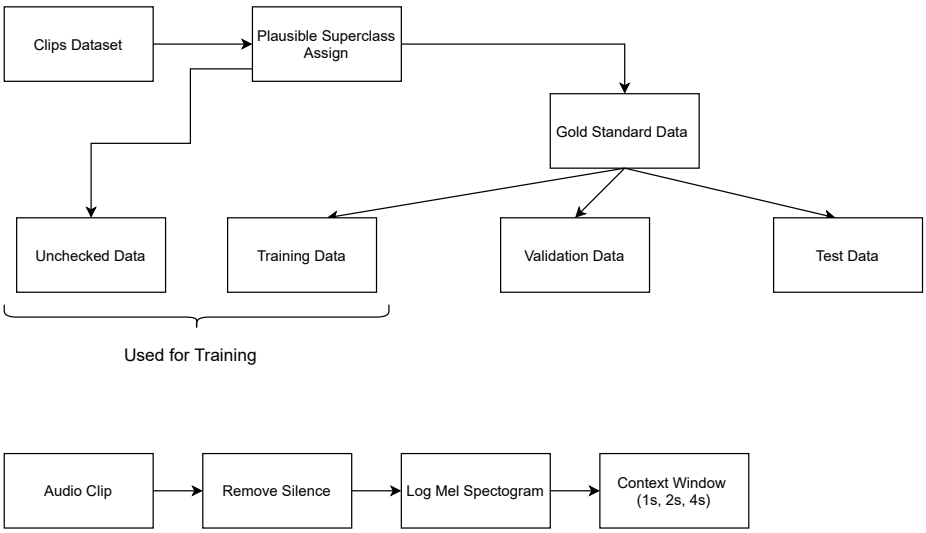

Figure 2: Data preparation and feature extraction pipeline

4.2 Evaluation Strategy

We randomly split the gold standard dataset into a training and a test set in a stratified manner. 35% of the clips were selected for the test set. Note that, we will use the unchecked clips with noisy labels in the future either for pretraining, or by refining their labels by some means. We randomly split the gold standard training dataset into 5 equally sized, stratified folds. We assigned only 1 of the folds as the validation set, and others were assigned to the "gold standard training" set. Alongside the gold standard training set, the unchecked were also used for training.

The same preprocessing and data preparation steps were followed for all dataset splits. Silent parts of the files were removed using power thresholding. We segmented the files to form either 1s, 2s or 4s clips with 50% overlap. The clips were framed into 25ms frames with 15ms overlap. 64 log mel filters were applied to each frame. The pipeline is shown in Fig. 2.

4.3 Optimization Strategy

In order to take advantage of the unchecked data while also not being biased by the noisy labels, we used a combination of pretraining on both unchecked and gold standard training data followed by finetuning on gold standard training data only.

We trained SwishNetV2 for 100 epochs with the AdamW optimizer [22] and a batch size of 256, 128 and 64 for 1s, 2s, 4s clips respectively on the combined training dataset (unchecked + gold standard training). The initial learning rate was set to 0.0005 and it was reduced by a factor of 0.5 if the validation loss didn't improve for 10 epochs. Finally, we finetuned the model on the gold standard training set for 10 epochs with a learning rate of 0.0005.

VGG16 trained really fast when initialized with ImageNet weights and reached convergence within 10 epochs with the SGD optimizer with learning rate of 0.001 and a momentum of 0.5, on the combined training dataset. Then it was finetuned on the gold standard training set for 1 epoch.

4.4 Results

the network structures fixed for all tests. Table 1 compares the sizes prediction speeds of the three networks as well as the baseline models. To simulate comparison of real-time computational cost, we fixed batch size to 1 for all models and compared their speed on CPU and GPU. Our CPU was Core i7 8700K, a 6 core processor and our GPU was NVIDIA GTX 1080ti and the networks were implemented in PyTorch [23].

Table 1: Network Sizes and Prediction Times per Sample

Network	No. of Parameters	Prediction Time (ms) per Sample for Different Sample Lengths						Weight file size
		1.0s		2.0s		4.0s		
		CPU	GPU	CPU	GPU	CPU	GPU	
SwishNet V2	15,127	2.15	1.85	2.52	1.87	2.86	1.87	59KB
VGG16	14.75M	31.4	2.92	51.6	3.97	89.7	5.88	56MB

Table 2: Overall Classification Accuracy (%) for Clips of Different Lengths. (GST = Trained on Gold Standard Training Set, GST + UT = Trained on Gold Standard and Unchecked Training Set, GST + UT + FT = Trained on Gold Standard and Unchecked Training Set, Fine Tuned on Gold Standard Training Set)

Network	1.0s	2.0s	4.0s
SwishNet (GST)	89.69%	91.50%	91.09%
SwishNet (GST + UT)	91.90%	93.42%	92.85%
SwishNet (GST + UT + FT)	92.45%	94.41%	94.54%
VGG16 (GST + UT + FT)	94.72%	96.01%	96.57%

We see that, similar to SwishNet, SwishNetV2 is fast on both CPU and GPU. SwishNet is approximately 10 times faster than VGG16 on CPU, due to its low parallelization requirement. Also in terms of number of pamameters and weight file size it is about 1000 times smaller than VGG16.

The classification results are presented in Table 2 and 3. We see that SwishNet performs very close to VGG16 in terms of accuracy, with the gap being only around 2%. The performance improves with increasing clip length. Also we see that the two stage of our training method by training on combined dataset and finetuning on gold standard dataset improves classification performance.

From the confusion matrices, we can see that recall for noise is low for both models, especially for smaller clip lengths, even though the speech/non-non-speech discrimination accuracy is high for all clip lengths. This result indicates that for smaller contextual windows, it is hard to differentiate noise from music. This is because, some of the noises, such as bells tolling or phones ringing are slightly

Table 3: Normalized Confusion Matrices for Clips of Different Lengths (Rows: True Labels, Columns: Predicted labels, Ordering: Speech, Music, and Noise. GST = Trained on Gold Standard Training Set, GST + UT = Trained on Gold Standard and Unchecked Training Set, GST + UT + FT = Trained on Gold Standard and Unchecked Training Set, Fine Tuned on Gold Standard Training Set)

Network	1.0s	2.0s	4.0s
SwishNet(GST)	$\begin{bmatrix} .90 & .05 & .05 \\ .02 & .92 & .06 \\ .02 & .11 & .86 \end{bmatrix}$	$\begin{bmatrix} .93 & .03 & .03 \\ .01 & .94 & .05 \\ .02 & .11 & .87 \end{bmatrix}$	$\begin{bmatrix} .93 & .03 & .03 \\ .01 & .94 & .05 \\ .03 & .11 & .86 \end{bmatrix}$
SwishNet(GST + UT)	$\begin{bmatrix} .97 & .01 & .02 \\ .03 & .91 & .06 \\ .06 & .05 & .89 \end{bmatrix}$	$\begin{bmatrix} .98 & .01 & .01 \\ .01 & .95 & .04 \\ .06 & .06 & 0.88 \end{bmatrix}$	$\begin{bmatrix} .96 & .03 & .01 \\ .00 & .97 & .03 \\ .06 & .08 & .86 \end{bmatrix}$
SwishNet(GST + UT + FT)	$\begin{bmatrix} .91 & .03 & .05 \\ .01 & .92 & .07 \\ .01 & .05 & .93 \end{bmatrix}$	$\begin{bmatrix} .96 & .01 & .03 \\ .01 & .95 & .03 \\ .02 & .06 & .92 \end{bmatrix}$	$\begin{bmatrix} .95 & .01 & .04 \\ .01 & .95 & .04 \\ .01 & .05 & .94 \end{bmatrix}$
VGG16 (GST + UT + FT)	$\begin{bmatrix} .94 & .01 & .05 \\ .01 & .94 & .05 \\ .01 & .03 & .95 \end{bmatrix}$	$\begin{bmatrix} .97 & .01 & .02 \\ .00 & .98 & .02 \\ .02 & .05 & .93 \end{bmatrix}$	$\begin{bmatrix} .98 & .00 & .02 \\ .00 & .97 & .03 \\ .01 & .03 & .96 \end{bmatrix}$

musical in nature and short segments of music may sound like noise. So, a longer context is necessary for discriminating noise from music.

5 Conclusion and Future Work

We manually added Gold Standard labels to AudioSet clips for speech, music and noise. We designed an improved version of SwishNet architecture – SwishNetV2 which is applied on log mel spectrum directly and achieved better accuracy. Also, we developed a scheme for training on noisy labels and fine tuning on gold standard dataset and showed that it improves accuracy. We showed that lean, fast CNN such as SwishNetV2 come very close in terms of accuracy to big 2D CNNs such as VGG16, even in such a diverse dataset as AudioSet.

In a future work we intend to include more unchecked data which should result in better performance. Also we intend to utilize unchecked data in a more sophisticated manner. One possibility is distillation from ensemble of models. We intend to add clips from other datasets into training. Possible candidates are FSD50K [24] and MUSAN [4]. Additionally, we intend to try and combine different features, including MFCC, log Mel spectrogram, and also NMF based features.

References

[1] Joe Wolfe. "Speech and music, acoustics and coding, and what music might be 'for'". In: *Proc. 7th International Conference on Music Perception and Cognition*. 2002, pp. 10–13.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436. ISSN: 1476-4687.

[3] Md Hussain, Mohammad Ariful Haque, et al. "Swishnet: A fast convolutional neural network for speech, music and noise classification and segmentation". In: *arXiv preprint arXiv:1812.00149* (2018).

[4] David Snyder, Guoguo Chen, and Daniel Povey. "Musan: A music, speech, and noise corpus". In: *arXiv preprint arXiv:1510.08484* (2015).

[5] Jort F Gemmeke et al. "Audio set: An ontology and human-labeled dataset for audio events". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 776–780.

[6] Savitha Srinivasan, Dragutin Petkovic, and Dulce Ponceleon. "Towards robust features for classifying audio in the CueVideo system". In: *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*. 1999, pp. 393–400.

[7] Zhu Liu, Yao Wang, and Tsuhan Chen. "Audio feature extraction and analysis for scene segmentation and classification". In: *Journal of VLSI signal processing systems for signal, image and video technology* 20.1 (1998), pp. 61–79.

[8] Lie Lu, Hao Jiang, and HongJiang Zhang. "A robust audio classification and segmentation method". In: *Proceedings of the ninth ACM international conference on Multimedia*. 2001, pp. 203–211.

[9] Silvia Pfeiffer, Stephan Fischer, and Wolfgang Effelsberg. "Automatic audio content analysis". In: *Proceedings of the fourth ACM international conference on Multimedia*. 1997, pp. 21–30.

[10] John Saunders. "Real-time discrimination of broadcast speech/music". In: *1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings*. Vol. 2. IEEE. 1996, pp. 993–996.

[11] Eric Scheirer and Malcolm Slaney. "Construction and evaluation of a robust multifeature speech/music discriminator". In: *1997 IEEE international conference on acoustics, speech, and signal processing*. Vol. 2. IEEE. 1997, pp. 1331–1334.

[12] Don Kimber, Lynn Wilcox, et al. "Acoustic segmentation for audio browsers". In: *Computing Science and Statistics* (1997), pp. 295–304.

[13] Tong Zhang and C-C Jay Kuo. "Video content parsing based on combined audio and visual information". In: *Multimedia Storage and Archiving Systems IV*. Vol. 3846. International Society for Optics and Photonics. 1999, pp. 78–89.

[14] Aaron Van Den Oord et al. "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499* (2016).

[15] Aaron van den Oord et al. “Conditional image generation with pixelcnn decoders”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 4790–4798.

[16] François Chollet. “Xception: Deep learning with depthwise separable convolutions”. In: *arXiv preprint* (2016).

[17] Tom Sercu et al. “Very deep multilingual convolutional neural networks for LVCSR”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4955–4959. ISBN: 1479999881.

[18] Shawn Hershey et al. “CNN architectures for large-scale audio classification”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135. ISBN: 1509041176.

[19] Naoya Takahashi et al. “Deep convolutional neural networks and data augmentation for acoustic event detection”. In: *arXiv preprint arXiv:1604.07160* (2016).

[20] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[21] Maxime Oquab et al. “Learning and transferring mid-level image representations using convolutional neural networks”. In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1717–1724. ISBN: 1479951188.

[22] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).

[23] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019), pp. 8026–8037.

[24] Eduardo Fonseca et al. “FSD50k: an open dataset of human-labeled sound events”. In: *arXiv preprint arXiv:2010.00475* (2020).