Karlsruhe Institute of Technology

Robust Voice Activity Detection in the Presence of Music using Neural Networks

Master's Thesis
by

Moritz Beeking

at the Department of Informatics
Institute for Anthropomatics and Robotics

First Reviewer:	Prof. Dr. Alexander Waibel
Second Reviewer:	Prof. Dr. Tamim Asfour
Advisor:	Dr. Sebastian Stüker

Process Period: April 15th 2020 – October 14th 2020

Ich versichere wahrheitsgemäß, die Arbeit selbständig verfasst und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, den 14.10.2020

Zusammenfassung

Voice activity detection (VAD) bezeichnet den Prozess, bei dem festgestellt wird, ob in einem Tonsignal menschliche Stimmen hörbar sind. VAD kann als Vorverarbeitungsschritt bei der automatischen Spracherkennung zum Einsatz kommen. Einen interessanten Einsatzbereich stellen die Tonspuren von Filmen dar. Auf Grund der oft vielfältigen Hintergrundgeräusche, insbesondere auch Hintergrundmusik, ist die Erkennung von Stimmen auf diesen schwierig. Gleichwohl könnte ein robustes VAD System Bestandteil eines zuverlässigen Systems zu automatischen Transkription und möglicherweise Untertitelerstellung sein.

Traditionell bildet die sorgfältige Zerlegung von Tonsignalen zur Erstellung komplexer Merkmalsvektoren die Grundlage von VAD Systemen. Obwohl Klassifikatoren auf Grundlage von Standardmerkmalen an Bedeutung gewonnen haben, stellt die Suche nach geeigneten Merkmalen noch immer einen aktiven Forschungszweig dar.
In dieser Arbeit werden gewöhnliche Mel-Frequenz Cepstralkoeffizienten (MFCCs) mit einigen exotischeren Merkmalen verglichen. Die besten Ergebnisse werden hierbei mit MFCCs erzielt, allerdings nicht unabhängig von den Hintergrundgeräuschen. Es wird gezeigt, dass 40-dimensionale MFCCs die besten Ergebnisse mit nichtmusikalischen Hintergrundgeräuschen liefern. Ist allerdings Hintergrundmusik vorhanden, führen 64-dimensionale MFCCs zu den besseren Ergebnissen.

Aktuelle VAD Systeme nutzen Neuronale Netze als Klassifikatoren. In dieser Arbeit werden ein System, welches ausschließlich aus bi-direktionalen long short-term memory (BLSTM) Schichten besteht, und ein System, welches aus einer Mischung aus frequency shifting time delay neural network (FSTDNN) und BLSTM Schichten besteht, verglichen. Letzteres beinhaltet hierbei so genannte gated FSTDNNs mit dilation. Dabei führt das kombinierte System zu geringfügig, besseren Ergbenissen. Die erzielten Ergebnisse sind konkurenzfähig zu anderen aktuellen Forschungsergebnissen.

Durch eine gründliche Erkundung von Eingabemerkmalen, neuronalen Architekturen und Parameterkombinationen wurde ein VAD System gefunden, das der Zielsetzung entsprach: Robust bei schwierigen Hintergrundgeräuschen, insbesondere in Gegenwart von Musik. Die Ergebnisse erlauben es, das System auf ein bestimmtes Ziel hin zu optimieren. Daher könnte das beschriebene System als Teil eines Spracherkennungssystems getestet werden.

Abstract

Voice activity detection (VAD) is the process of determining, whether human voice can be heard in an audio stream. VAD can be used as a preprocessing step for automatic speech recognition. One interesting source of audio recordings are the audio tracks of movies. These pose a particularly difficult challenge, because of the high level and diversity of background noise, including music. Robust voice activity detection could be one part of a reliable automatic transcription or subtitle generation system. This makes searching for a robust VAD system a challenging but worthwhile pursuit.

Traditionally, careful decomposition of the audio signal, yielding complex acoustic features, was the foundation of VAD. Although classifiers based on standard features have become more important, suitable input features are an area of active research. This work compares standard Mel-frequency cepstral coefficients (MFCCs) with some more exotic features. MFCCs show the best performance, but not independent of background noise. For non-musical background noise, 40-dimensional MFCCs are found to best detect voice activity. However, in the presence of background music, 64-dimensional MFCCs yield better results.

Contemporary VAD systems use neural networks as classifiers. In this work, a system consisting of solely bi-directional long short-term memory network (BLSTM) layers, and a system consisting of a combination of frequency shifting time delay neural network (FSTDNN) and BLSTM layers are compared. The latter incorporates so called gated FSTDNNs with dilation. The combined system yields the better performance, even if only by a small margin. The reported results are on par with contemporary research.

A thorough exploration of input features, neural network architectures and parameter combination yielded a VAD system that met the requirements: Robust under difficult background noise conditions and especially so when music is present. The results allow for the system to be optimized for a chosen task. Thus, the proposed system could be tested as part of an automatic speech recognition system.

Contents

List of Tables

List of Figures

1 Introduction

Voice activity detection (VAD) is the process of detecting whether one or more human voices can be heard within an audio stream. It differs from speech activity detection (SAD) slightly as the latter requires the heard voices to produce intelligible speech, while the former also applies to non-speech articulations like sighs[HSN19]. However, one has to be careful as this fine difference is not always adhered to in literature. Speech/music discrimination on the other hand is the process of differentiating whether a given recording or frame from an audio signal contains (primarily) speech or music. This work proposes a VAD system capable of detecting not only voice activity but also differentiating whether there's noise or specifically music in the background.

1.1 Motivation

VAD can be used as part of systems for voice transmission, speech enhancement, automatic speech recognition (ASR) and related applications. This work focuses on VAD in media, especially audio tracks of movies. These are particularly difficult, as background noise is common and varies heavily. Background noise may include car and machinery noises, nature sounds, even gunshots and a myriad of different other sounds depending on the plot of the movie. Such a variety poses a difficult problem when trying to estimate the attributes of the encountered noise in the chosen feature space [HSN19]. Another challenge is the omnipresent background music. While discrimination of music without vocals from speech can be achieved by using carefully selected features [SS97], music with vocals is very similar to speech and requires more sophisticated approaches [CG01]. Despite the described difficulties the research on VAD under such circumstances is worthwhile. The archives of television stations all around the world are filled with almost a century worth of voiced movies. Unfortunately, for most of them no transcripts exist. This hurts the possibility to search them by topics properly, or to generate subtitles. The generation of subtitles is especially interesting as those could be reasonably translated to make these archives of human culture available to people outside the sprachraum of the original production. Notably, this is not limited to historic movies, but extends to lots of smaller contemporary productions lacking the means for subtitling and translation. Because of the sheer amount of movies, annotating voice activity by hand is not feasible. Thus a VAD system which is robust under the movie specific circumstances is required.

1.2 Approach

The general approach to VAD is to first split an audio stream into reasonably sized frames for preprocessing. Subsequently fitting features are extracted from these frames. These are fed into a classifier for the actual VAD decision. Those decisions may be performed per recording or per frame of limited duration. The output might either be a label per recording or a label per given time range within in the recording. Usually 10 ms are the smallest used range, being the duration of a typical extracted audio frame [HSN19; EWSS13]. Longer spans are used but less common[LWS15]. Another possible output are the endpoints of voice activity[CLS+17].

This work is aimed at performing VAD on a frame-wise basis, providing either these labels or endpoints as output during inference. Strictly speaking, the aim of this work is an extension of the VAD problem. VAD is a binary classification problem whether voices can be heard or not. This work aims at further annotating the nature of background noise, discriminating between music and other noises as well as clean speech. When designing a VAD system there are three important considerations: First, which features to use. This was the main area of research for early VAD systems and still is an actively pursued topic. Carefully handcrafting features is one possibility, [CWW14] lists several contemporarily used features. Another possibility is to directly learn features from raw waveforms [ZSSP16]. Recently, the most common approach is using standard features like Mel-frequency cepstral coefficients (MFCCs) and a more complex Classifier [HSN19; KSI18; GG18]. The proposed system also uses MFCCs. The multi-resolution cochleagrams (MRCGs) [CWW14] introduced 2014 are used for comparison.

The second consideration is how to classify a given feature vector. While a lot of different classifiers have been proposed, most contemporary work focus on NNs. A NN based on BLSTMs is thoroughly explored in this work. Different numbers of layers and sizes of hidden layers are examined. Further analysis is conducted in search of the optimal scheduler during training and the best way to combine the directions of the BLSTM. Furthermore, following [LMHK20], an attempt was made to incorporate an attention mechanism into the system. The WebRTC VAD [Goo20] is used as a baseline for the VAD performance of the proposed systems. Notably, the system proposed in this work is not a "pure" VAD system, as it is not a binary classifier but knows four possible classes.

The third consideration is which data should be used for training, development and testing. The AVA-corpus [CRE+18] is well suited to develop and test a system addressing the problem described above. It features four classes: clean speech, speech with music, speech with noise and no speech. Different smoothing techniques were tested as well.

1.3 Overview

The remainder of this work is split into eight chapters. Chapter 2 describes the concepts and technologies used in this work. Subsequently, some corpora of interest to the areas of VAD and SAD are outlined in chapter 3. Their usefulness to this

work is evaluated and the decision to mainly use the AVA-speech corpus is justified. Chapter 4 starts with a short history of VAD and speech/music discrimination before providing an overview on some contemporary approaches to VAD. In chapter 5, the systems proposed in this work will be described in theory. Some information on used technologies and software is also located there. The conducted experiments are illustrated in chapter 6 and the results discussed in chapter 7. Finally, chapter 8 will wrap up this work and give an outlook on the most promising possible continuations of this work. A glossary and list of acronyms are appended.

2 Fundamentals

With the general idea and purpose of voice activity detection (VAD) systems covered, this section will outline the necessary fundamentals for a contemporary VAD system. The descriptions follow the flow of data in a VAD system. First, the preprocessing and feature extraction are described, followed by the structure of the classifiers. Afterwards, the training process for the classifiers is described. The post processing steps are outlined next, and the chapter is ended with a description of the metrics used for evaluation.

The recording as the step before the preprocessing will not be covered here. This is due to the fact that voice activity detection in the presence of music, the ultimate goal of this work, is primarily aimed at media audio. Therefore the recording settings are diverse and in no way controlled by the system or its user.

2.1 Preprocessing

The first step for performing VAD and related tasks is the preprocessing of the input audio signal. Preprocessing towards voice activity detection can be approached in different ways. Some historical methods and notable techniques from recent works mentioned in chapter 4. For the predominant neural network based voice activity detectors the most frequently used inputs are Mel-frequency cepstral coefficients (MFCCs). Recently multi-resolution cochleagrams (MRCGs) as introduced by [CWW14] led to some promising results. Both are used in the systems proposed in this work and thus outlined subsequently.

2.1.1 Mel-frequency cepstral coefficients

MFCCs have been around for over forty years [DM80] and are regularly described in textbooks [KLW19, pp. 121 sq.], [SDD19, pp. 54 sqq.]. Descriptions differ slightly, yet the general procedure remains the same. The following explanation is aimed at being consistent with the implementation. A depiction (fig. 2.1) is also enclosed.

Humans do not actively perceive the phase of a signal [PNHR88]. Therefore, after sampling and quantization, the first step of preprocessing usually is to transform the signal to a representation in the frequency domain[KLW19, p. 371]. For MFCCs, as for different other features, this is achieved by performing a discrete fourier transform (DFT) over short windowed excerpts of a signal [SS12]. The windows should be short enough to reasonably assume the signal to be stationary within [KLW19, p. 373],

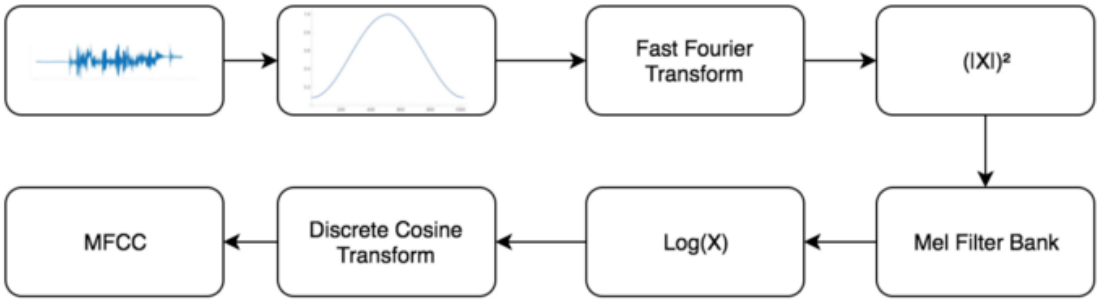

Figure 2.1: Procedure to generate MFCCs from signal waveform [Cho18]

typically 10 ms to 30 ms [SDD19, p. 54]. To extract the windowed excerpts, called frames, the signal waveform is multiplied by a sliding window function. Hann or Hamming windows, two closely related cosine-sum windows, are the most commonly used window functions for this purpose [KLW19, p. 373].

The spectrogram is derived from the squared magnitude of the DFT. The resulting spectrogram is then passed through several Mel scaled triangular filters. The Mel scale is aimed at measuring the different frequencies as perceived by humans rather than equally spacing the filters according to the actual physical frequencies. [KLW19, p. 372-74]

Subsequently the Mel filter bank log energy (MFLE) is calculated for each filter. To retrieve the actual MFCCs a linear transformation of the MFLEs is needed to increase robustness and decorrelate the features. The most commonly used is the discrete cosine transform. The resulting cepstrum's amplitudes are the MFCCs. [SS12]

Details concerning the actually chosen parameters can be found in subsection 5.3.1.

2.1.2 Multi-resolution Cochleagrams

Multi-resolution cochleagrams are a relatively recent feature introduced in [CWW14]. The composition, along with some standard parameters, are depicted in fig. 2.2.

Multi-resolution cochleagrams are based on cochleagrams as described by [PNHR88]. Those are psychoacustically motivated representations of a signal. They are based on gammatone filters and spaced according to the equivalent rectangular bandwidth (ERB). The ERB is a coarse approximation of the sensitivity of the cochlea to different frequencies. The cochleagram itself is aimed at representing the movement of the basilar membrane. [PNHR88; CWW14]

Multi-resolution cochleagrams are aimed at encoding the context of an audio frame directly into a single feature matrix. Just as for MFCCs the frames are windowed excerpts, but for the cochleagrams they are chosen to have a slightly shorter duration. The MRCGs consist of four 64-dimensional vectors. First, cochleagrams with a window size of 20 ms and a shift of 10 ms. The second vector also consists of cochleagrams, but derived from significantly longer frames of 200 ms. The shift

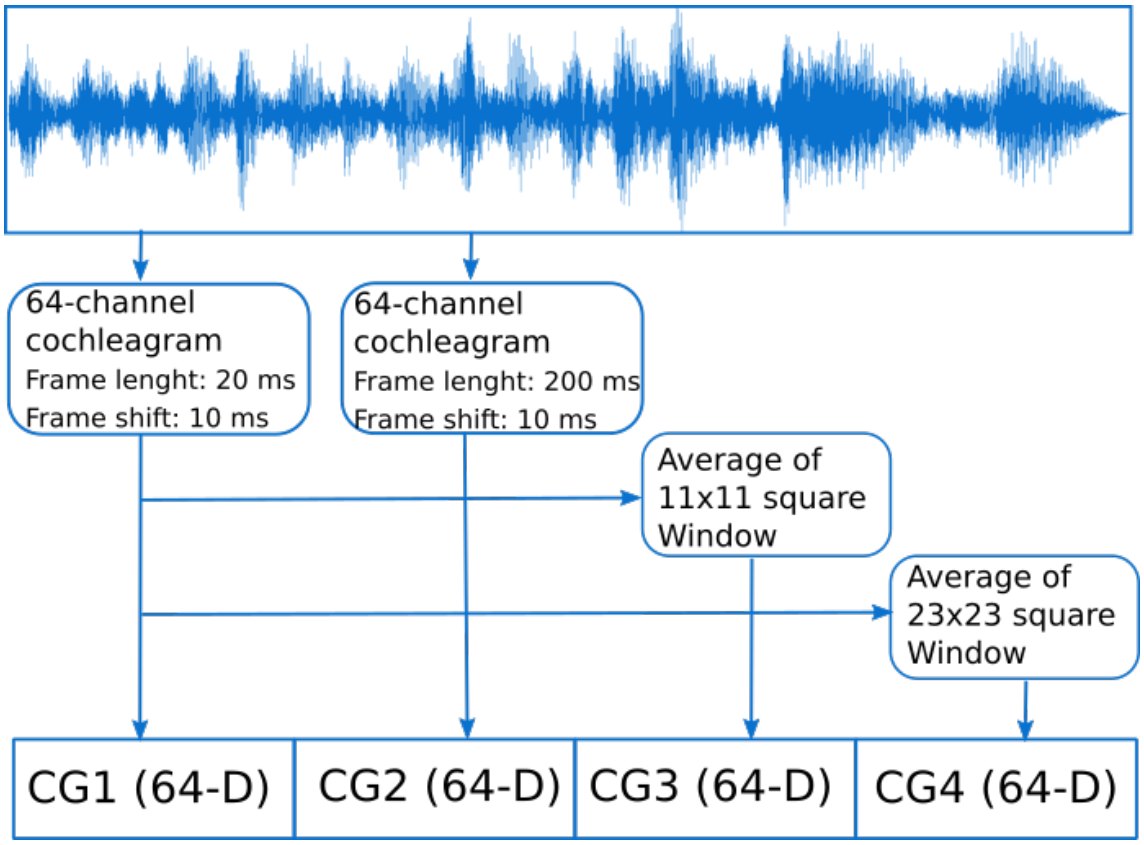

Figure 2.2: Procedure to generate MRCGs from signal waveform

between the frames stays the same, so the number of derived vectors along a recording is also identical. The third vector consists of the averages of a square window of size 11x11 across the first cochleagram and those of the neighboring frames. Zero padding is used at the lower and upper frequency borders of the cochleagrams as well as on the temporal borders of the window. The fourth vector is constructed in the same way as the third, but with a window size of 23x23. [CWW14]

2.2 Neural Networks

A neural network is composed of small processing units, often called neurons, and weighted connections between them [Gra12, p. 13]. The naming derives from the history of neural networks (NNs) as approximations for the inner workings of the brain. Although NNs are in most cases not aimed at explicitly imitating the brain anymore, the names stuck [GBC16, p. 169].

Each processing unit has an activation function, which determines the output of that unit given the weighted sum of the inputs. Because “any combination of linear operators is itself a linear operator” [Gra12, p.16], nonlinear activation functions like the sigmoid (σ) or hyperbolic tangent (tanh) function (eq. (2.1)) are used. These are also differentiable, an important property to be trainable using gradient decent [Gra12, p. 16]. Gradient descent is described in section 2.3.1. Because there is a linear transform between σ and tanh any function that can be approximated by a

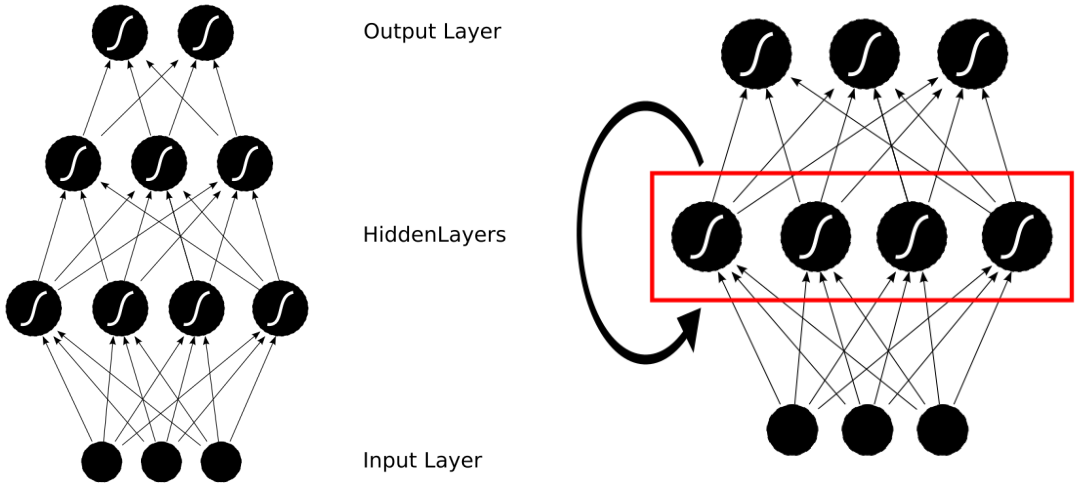

Figure 2.3: Feedforward and recurrent NN [Gra12, p. 14/21]

NN with sigmoid activation functions can also be approximated by a NN with hyperbolic tangent activation functions [Gra12, p. 15]. NNs are usually organized in so-called layers. Although in theory a single layer with sufficient units can approximate any function, in practice multilayered architecture need far less parameters for sufficiently exact approximations [GBC16, p. 198].

$$\begin{aligned} \sigma(x) &= \frac{1}{1+e^{-x}} \\ \tanh(x) &= \frac{e^{2x}-1}{e^{2x}+1} \\ \implies \tanh(x) &= 2\sigma(2x)-1 \end{aligned} \tag{2.1}$$

2.2.1 Feedforward Network

A network with connections only from one layer to the next is called feedforward network [Gra12, p. 14]. Such a network is depicted in the left half of fig. 2.3. For historical reasons such networks are sometimes called multilayer perceptrons (MLPs). Depending on the number of hidden layers, such a network might also be called “deep”. However, the number of hidden layers to consider a network “deep” is not properly defined and subjected to change [KLW19, p. 154].

2.2.2 Softmax

The softmax function can be used to convert the output activation of a classifier into a probability distribution. It is defined as

$$softmax(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.2}$$

with z being a vector consisting of one real value per class. After applying the softmax function, the values in the resulting vector are all between 0 and 1 and add up to one, i.e. they resemble a probability distribution. The softmax function is also derivable. Therefore it is often used as a final layer in a NN. [GBC16, pp. 184–187]

2.2.3 Time Delay Neural Networks

Time delay neural networks (TDNNs) were introduced in [WHH+89] for phoneme recognition. The idea is to train a network to recognize patterns useful to the task at hand and then shift it over the input along the time axis.

Every step each of the hidden units is fed with the activation of the input layer from the current or a previous time step. Another interpretation of this architecture is a weight pattern being convolved with the contents of a sliding window over the input. Thus, it is possible to achieve time invariance and an efficient incorporation of context information. [LWH90]

The weight patterns are often called "filters". Originally, they were moved exactly one step at a time. However, moving them further is possible, and the corresponding parameter is called the "stride" If there are gaps between the inputs of a filter, the network is called "dilated". The "dilation" then is the number of steps between each input. Thus, a dilation of one denotes a non-dilated network, a dilation of two gaps of one input step and so on. [KLW19, pp. 264 sq.]

TDNN based architectures are widely used for VAD systems, often arranged in several layers [CLS+18; War17; SSVC17]. If the filter kernel of a TDNN is shifted not only along the time, but also along the frequency axis, the corresponding network is considered a frequency shifting time delay neural network (FSTDNN). Introduced by [IW90], these systems are widely used for VAD applications [HSN19; ZSSP16; CRE+18], especially when little to no preprocessing is conducted.

The input to a TDNN can be padded. Padding describes the process of adding additional values at the borders of the input. The most common approach is to add zeroes, consequently called zero-padding. The motivation is as follows: Consider an input of width W, a filter with I inputs, stride S and dilation D, and a padding P. The width of the output N can then be calculated as

$$N = \frac{W - (F * D) + 2P}{S} + 1 \tag{2.3}$$

Thus, padding is a convenient way to trim the size of the output of a TDNN. [KLW19, p. 264]

2.2.4 Long Short-Term Memory

Long short-term memory networks (LSTMs) were first introduced by [HS97]. They are a type of recurrent neural networks (RNNs), depicted in the right half of fig. 2.3. Because of the recurrent connections, RNNs are capable of encoding previously seen information in the state of the NN itself [Gra12, p. 20]. Thus, temporal dependencies in sequences of arbitrary length can be exploited for predictions [GBC16, pp. 373 sq.].

However, with each time step especially training gets more difficult because of the so called "vanishing gradient problem" [Gra12, p. 32]. This results from the properties of the backpropagation through time algorithm. For details on that algorithm see section 2.3.1. The problem however is, that with increasingly long training sequences the gradient of the error resulting from wrong predictions gets very small, as it

is backpropagated through the network. Because the weights of the network are adjusted depending on said gradient, the learning process becomes extremely slow. [Hoc98]

One solution to this problem is the LSTM architecture. A single LSTM cell is depicted in fig. 2.4. The distinctive feature are the three gates. The input gate controls how much of the input is added to the cell's internal state in each time step. The forget gate controls how much of the previous state is preserved. The output gate controls to what extend the current state is propagated. The net input denotes the flow of information from the input of the network or the layer below the given unit into it. The net output denotes the flow towards the subsequent layer or the output of the network. All three gates calculate their activation from weighted sums over the current input, the previous output and the previous cell state. As for other NNs, these weights are the learnable parameters of the system. Because the gates control the change of the activation of a given unit, each cell is capable of preserving an activation and its gradient. Thus, the vanishing gradient problem is much less of a concern. [Gra12, pp. 33 sq.]

A special case of LSTMs are bi-directional long short-term memory networks (BLSTMs). Standard RNNs process the given sequence in the order in which it is presented. Thus, they only make use of the past, also called right, context of a frame for the corresponding prediction. However, if the left, also called future, context is already known at computation time, it can be utilized as well. [GBC16, pp. 394 sq.]

To make use of both past and future context, two RNNs are used. One processes the sequence in the forward direction, starting with the first frame. The other processes the same sequence in the inverted, backward direction. Both are connected to the same subsequent layer. Bi-directional RNNs (BRNNs) have been found to consistently outperform their unidirectional counterparts, thus only BLSTMs will be used in the proposed system. [Gra12, pp. 22 sq.]

2.3 Training

The process of fitting the weights in a NN to a specific problem is called training. The weights are adjusted using the gradient descent and the backpropagation algorithms. Too achieve this, an error function, representing how bad the current model represents the training data is needed. In this work, the cross-entropy loss is used. Schedulers like Adam or squared mean over root mean squared cubed (SMORMS3) can be used to control the speed of the training process. The remainder of this subsection describes the mentioned techniques in detail. [GBC16, pp. 374 sqq.][KLW19, pp. 41 sqq.][Gra12, pp. 4 sqq.]

2.3.1 Gradient Descent

The most popular approach to optimize NNs is gradient descent [GBC16, pp. 151 sq.]. Therefore, an objective function (see section 2.3.3) is derived with respect to the network weights. During training, the weights are then adjusted slightly in the directions of the negative slope of the gradient. [Gra12, p. 18]

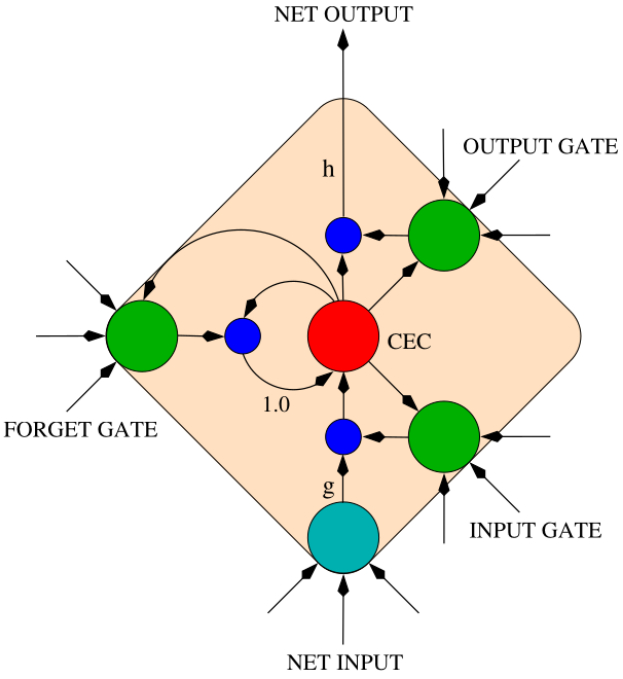

Figure 2.4: A single LSTM cell [Gra12, p. 34].
The small blue circles represent multiplicative units. The big red circle represents the constant error carousel (CEL). The activation of the cell is preserved via the self connection of the CEC. The forget gate is linked to this self connection as to dampen activation considered not useful. [HS97]

The classical gradient descent algorithm calculates the error gradient on the entire training data. However, modern training sets tend to consist of lots of training samples. Thus, calculating the predictions and the resulting error of the entire set is time consuming. To keep the training reasonably fast, the weights are updated after a part of the training data has been consumed. This is possible because the gradient calculated from a reasonably large subset of the training data, drawn at random, is a sufficient approximation of the gradient calculated from the whole set. This technique is known as online or stochastic gradient descent (SGD). The subsets of the training data are commonly called minibatches. [Gra12, p. 18] [GBC16, pp. 151 sq.]

2.3.2 Backpropagation

A neural network generates its output by first calculating an activation in every unit of the input layer. This activation is then used as input to the following layer, and so on until the output layer calculates the output. This process is called forward propagation. In the same way the error derived from the objective function can be propagated from the output layer towards the input. This process is called backpropagation. The error with respect to each individual weight can be calculated by repeatedly applying the chain rule. Thus, the weights of a multi-layer NN can be adjusted effectively. [GBC16, p. 294] [RHW86]

The training of RNNs is a special case. Because of the recurrent connections, the activation and the gradient of a unit depend not only on the current input, but also on the previous input. To train such networks an algorithm called "backpropagation through time" is used. The recurrent network is treated like a very deep network with weight sharing. This representation is called the unrolled computational graph. Along this graph, the gradient can be propagated. [GBC16, pp. 384 sqq.]

2.3.3 Cross-entropy Loss

One error function fit for gradient descent is cross-entropy loss [GBC16, p. 226]. Sometimes it is also, more descriptively, called the negative log likelihood [KLW19, p. 162].

Let x be an input frame, $q(x)$ the corresponding label, represented as a one-hot encoded vector, and $p(x)$ the corresponding prediction. In the multinomial case, both $q(x)$ and $p(x)$ have one entry per possible class $c \in C$. The cross-entropy loss for all samples $x \in X$ can then be calculated with eq. (2.4). In the binary case, having two outputs is not necessary. As the output of a classifier should be a probability distribution, it is sufficient to provide one probability $p(x)$, the other then is $1 - p(x)$. In that case the label $q(x)$ can be represented as either 1 or 0. Thus, the second sum becomes superfluous and eq. (2.5) is sufficient as objective function.

$$CEL_{mul} = -\sum_{x \in X} \sum_{c \in C} q(x) \ln p(x) \tag{2.4}$$

$$CEL_{bin} = -\sum_{x \in X} q(x) \ln p(x) \tag{2.5}$$

2.3.4 Schedulers

One important consideration in SGD is how to choose the step size by which the weights are adjusted. It is necessary to adapt this value to achieve convergence reasonably fast [GBC16, p. 294]. An algorithm used to control this value is called a training scheduler.

Adam [KB17] is one very popular scheduler. Because the gradient is calculated with respect to each individual weight, an individual stepsize may be chosen as well. Adam does so by keeping track of the previous adjustments of the weights. Thus, a momentum for each individual weight can be calculated, avoiding oscillations. [GBC16, pp. 308 sq.] [KB17]

Another, similar algorithm is SMORMS3 [Sim15]. Like Adam, it is based on RMSProp [GBC16, p. 308]. SMORMS3s main assumption is, that although the mean of the gradient approaches zero when close to a minimum, the noise need not to. Additionally, the learning rate is tied to the denoising factor to keep it "the (known) scale of the parameters themselves than in the (relatively unknown) scale of the gradients" [Sim15].

2.4 Postprocessing

The output of the classifier consists of class probabilities for each class and time step. One goal of postprocessing is to incorporate knowledge not properly learned by the classifier into the final decision. For example, some kind of smoothing, keeping the final decision from oscillating between classes, can be useful. Another goal of postprocessing is tailoring the output towards the subsequent tasks. For automatic speech recognition (ASR) some superfluous noise frames are tolerable. Clipping on the other hand is problematic. Therefore, a threshold shifting the output to either under- or overestimate the presence of voice activity can be useful as well. Subsequently, the tested postprocessing techniques are outlined.

2.4.1 Padding

All smoothing techniques discussed in this section rely on a smoothing window. An important consideration when working with windows is how the algorithm should behave at the borders of the input. Otherwise the smoothed output might be distorted towards the borders or consist of less values. One way to avoid this is to pad the input with additional values. Zero padding, i.e. adding a fitting number of zeroes on both sides preserves dimensionality, but may still distort the result. The alternative used in this work is called "mirror" or "reflect" padding. The padded values are mirrored versions of the input on the other side of the original border. The left and right padding are denoted in eq. (2.6). w is the context width, i.e. the width of the context window to both sides of the current frame. n is the length of the input. The padded input is the concatenation of left pad, original input and right pad.

$$\begin{aligned} \text{left pad: } & p(x_{-t}) = p(x_t), t \in [1..w] \\ \text{right pad: } & p(x_{n+t}) = p(x_{n-t}), t \in [1..w] \end{aligned} \tag{2.6}$$

2.4.2 Continuous Smoothing

One straightforward approach to smoothing is mean-smoothing the class probabilities along the time axis. This is achieved by replacing each class probability by the result of a multiplication of itself and its neighboring frames with a window function as denoted in eq. (2.7). A hamming window [SDD19, p. 46] was used in this work and is assumed in the formula. w is the context width, i.e. the number of frames to both sides of the current frame to be considered during smoothing. $W = 2w + 1$ is the total width of the smoothing window, including the central frame. x_t is considered the current frame, $p(x_t)$ its unsmoothed probability and $\hat{p}(x_t)$ the smoothed probability. Each probability is multiplied by a corresponding entry of the smoothing window. The sum is then divided by the sum of the smoothing window alone. This is important because if the general magnitude changed, the output would no longer resemble a probability distribution over the possible classes in each frame.

$$\hat{p}(x) = \frac{\sum_{n=0}^{n=W} 0.54 - 0.46 \cos \frac{2\pi n}{W-1} p(x_{t-w+n})}{\sum_{n=0}^{n=W} 0.54 - 0.46 \cos \frac{2\pi w}{W-1}} \tag{2.7}$$

2.4.3 Thresholding

During training some loss function, in this work the cross-entropy loss (CEL), between the ground truth and the prediction of the network is minimized. The resulting predictions do not necessarily yield the best results for the subsequent tasks. For example, in ASR scenarios a few noise frames handed over to the ASR system are tolerable. Clipping on the other hand should be avoided, as it may harm recognition performance severely. A common measure to shift the output of a VAD system towards a wanted behavior is thresholding. Based on the results on the development data, a threshold τ when to consider a frame to contain voice activity can be fixed. The output class c is then assigned according to eq. (2.8). If there is more than one class representing voice activity, as is the case for the AVA corpus, the decision becomes a little bit more complicated. However, as long as the threshold is only used to shift between classes with or without voice activity and there is only one class representing frames without voice activity, the changes are minimal. The resulting formula is eq. (2.9). $\hat{p}(x)_k$ is the smoothed probability of frame x belonging to the class with index k. V is the set of indices belonging to classes representing voice activity, n is the index of the class representing no voice activity. So, if the probability of the frame containing no voice activity is below the threshold, the most probable class representing voice activity is assigned to the frame. Otherwise the class representing no voice activity is assigned to the frame.

$$c(x) = \begin{cases} \text{voice activity}, & \text{if } p(\hat{x}) \geq \tau \\ \text{no voice activity}, & \text{otherwise} \end{cases} \tag{2.8}$$

$$c(x) = \begin{cases} c[argmax_i \hat{p}(x)_{i \in V}], & \text{if } p(\hat{x})_n \leq \tau \\ \text{no voice activity}, & \text{otherwise} \end{cases} \tag{2.9}$$

2.4.4 Max-vote Smoothing

After a hard decision towards classes was made, single frames with or without voice activity in otherwise different sequences are unlikely to be correct predictions. One simple way of correcting these frames is max-vote smoothing. Within a window of a given length, any frame not belonging to the predominant class is considered an erroneous prediction and the class of the surrounding frames is assigned instead.

2.4.5 Length-based discrete smoothing

By analysis of the training data a minimum length of sequences with and without voice activity to be expected can be be derived. When these lengths are known, a postprocessing step eliminating shorter sequences can be used. Any elimination of a sequence leads to the combination of the surrounding sequences into a larger one. Therefore, the smoothing process is started by assigning single frames surrounded by frames of a different class to that class. This is repeated for sequences of length two and so on until the minimum length is reached. It is possible to perform this with different minimum lengths per class.

2.5 Metrics

This section describes the metrics used during evaluation of the proposed systems. Throughout the following subsections let M be the size of the entire dataset. Let P be the number of samples belonging to the positive class, in this case voice activity. Note that several of the metrics below are aimed at binary classifiers. For using them with the multinomial classifier used, all classes representing voice activity were considered equal. Let further N be the number of samples belonging to the negative class, in this case no voice activity. T_p and T_n, denote the total numbers of samples correctly assigned to the positive or negative class respectively. F_p and F_n denote the corresponding values for incorrectly assigned samples.

2.5.1 Accuracy

Accuracy is a measure of the correctly classified proportion of the dataset [GBC16, pp. 103 sq.]. The calculation is denoted in eq. (2.10). Accuracy is a good measure of performance for balanced datasets, but suffers from class imbalance. For example, a model assigning the negative class to every sample has an accuracy equivalent to $\frac{N}{S}$, which might be a value close to one.

$$\text{Acc} = \frac{T_p + T_n}{M} \tag{2.10}$$

2.5.2 Precision

Precision is a measure of how many of the predictions of a frame belonging to the positive class are correct. The calculation is as follows:

$$\text{Prec} = \frac{T_p}{T_p + F_p} \tag{2.11}$$

2.5.3 Recall

Recall is a measure of how many samples belonging to the positive class are correctly classified [Cho20, p. 559]. The calculation is denoted in eq. (2.12). Recall is of special interest for VAD systems aimed at speech recognition, as it is a good approximation of the level of clipping.

$$\text{Rec} = \frac{T_p}{M} \tag{2.12}$$

2.5.4 F1-Score

The F1-Score is the harmonic mean of the precision and the recall [Sas07]. The calculation is as follows:

$$\text{F1} = 2 * \frac{\text{Prec} * \text{Rec}}{\text{Prec} + \text{Rec}} \tag{2.13}$$

2.5.5 Error Rates

The following error rates are often reported in VAD literature. The frame error rate is usually not reported, as it can easily be derived from the commonly reported accuracy. The other three were especially reported for the movies of the Hollywood corpus by [LWS15] and [JASJ17] and thus used for evaluation of the proposed systems.

False Positive Rate

The false positive rate is the rate, at which a sample belonging to the negative class is assigned to the positive class. Its calculation is as follows:

$$\text{FPR} = \frac{F_p}{N} \tag{2.14}$$

False Negative Rate

The false negative rate is the rate, at which a sample belonging to the positive class is assigned to the negative class. Its calculation is as follows:

$$\text{FPR} = \frac{F_n}{P} \tag{2.15}$$

Equal Error Rate

The equal error rate is the value of false positive and false negative rate, if the threshold is adjusted such that the both are equal. Calculating the equal error rate and the corresponding threshold on the training data is necessary to optimize a system towards the F1-Score.

Frame Error Rate

The frame error rate is the proportion of the samples assigned to the wrong class. Thus, it is the opposite of the accuracy. The calculation is as follows:

$$\text{FER} = \frac{F_p + F_n}{M} = 1 - Acc \tag{2.16}$$

2.5.6 Receiver Operating Characteristics Curve

Receiver operating characteristic (ROC) curves are useful for visualizing the performance of a binary classifier. A ROC curve of one of the proposed systems is shown in fig. 2.5. To understand it, picture the following: Initialize a system with a threshold of one. Thus every sample, regardless of the networks output, is placed in the negative class. The TPR is zero, the FPR likewise. This corresponds to the point $(0, 0)$ in the lower left. As the threshold is moved towards zero, more and more

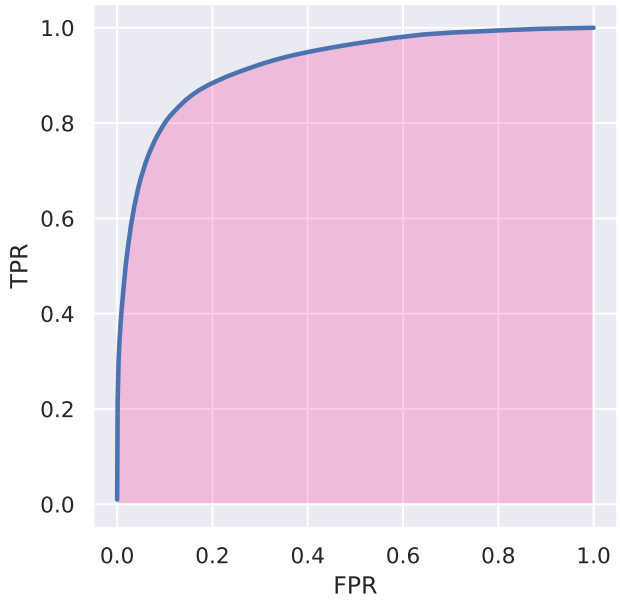

Figure 2.5: ROC curve of one of the proposed systems.

samples are assigned to the positive class. This leads to an increase of the TPR, moving the curve in y-direction, and also an increase in FPR, moving the curve in x-direction. As the threshold reaches zero, all samples are placed in the positive class. TPR and FPR are both one, corresponding to point $(1, 1)$ in the upper right. A classifier is desired to assign samples to their corresponding classes, yielding a low FPR and high TPR. Thus a threshold yielding an ROC value as far "northwest" as possible is desirable. [Faw06]

Another metric related to the ROC curve is the area under the curve (AUC). In fig. 2.5 the lightly pink colored area represents the AUC. The AUC is theoretically between zero and one. Yet only values above 0.5 are relevant, as that is the AUC for randomly assigning classes. Notably, the AUC is equivalent to the probability of the classifier of assigning a higher probability to a sample belonging to the positive class than to one belonging to the negative class. [Faw06]

3 Corpora

In the context of VAD, and related tasks like ASR, a *corpus* is a collection of input data and some accompanying annotations. For VAD a corpus should consist of audio files with annotation indicating whether voices can be heard. Generally interesting attributes of a corpus are its size as well as the assumed precision of annotations. The first consideration naturally is whether the annotated classes match the task at hand. As this work focuses on the difficulties of separating music from speech, the presence of music should also be annotated. These considerations lead to a relatively sparse selection of useful corpora, which will be described subsequently. For further reference some corpora linked to either speech/music discrimination or VAD but not used for this work are also listed.

3.1 Used Corpora

Ultimately this work used only two corpora. The training was exclusively done on the AVA-corpus described subsequently. Because the AVA-corpus is not separated into training and test data, it could not be used to evaluate the performance in comparison with other works. For this, a corpus of four Hollywood movies, simply called Hollywood corpus throughout this work, was used.

3.1.1 AVA-Speech

The AVA-Speech corpus was introduced by a team of google researchers in 2018 as „A Densely Labeled Dataset of Speech Activity in Movies “[CRE+18]. It features about 40 h of movie excerpts taken from 160 movies publicly available on Youtube. Each of the excerpts is between the fifteenth and thirtieth minute of the corresponding video. There are four possible labels: “Speech and Noise”, “Speech and Music”, “Clean Speech”, and “No Speech”. The annotation process is described as follows: When no clear distinction was possible, the labels were to be considered in the aforementioned order. The 15 min excerpts were split into 1 min clips. Each clip was annotated by three humans with the final labels chosen by majority vote. The labels were aligned to frames of 10 ms.
Movies as a source for a voice activity detection corpus offer several advantages. Because the movies originate from different countries, several languages from different continents are present within the corpus, allowing a system to be trained on general properties of speech rather than on specifics of a certain language. Furthermore different situations like monologues, dialogues of two or more persons, and a variety

Table 3.1: Movies in the Hollywood movie corpus [LWS15]

Movie title	Genre	Duration	Speech in %
I am Legend	Action Thriller	1:40:22	18.3
Kill Bill: Volume 1	Martial Arts	1:46:08	19.2
Saving Private Ryan	War	2:42:27	32.1
The Bourne Identity	Action Thriller	1:58:24	26.7
Total	-	8:07:21	25.2

of background noises are present in the corpus. Especially when considering voice activity detection with music in the background movies are very useful, as they can provide a much more realistic mixture of speech and music than artificially created training data from clean speech and music.
[CRE+18] offers three baseline models for comparison. The voice activity detector of WebRTC is their baseline with two different, neural network based, superior models. This corpus was primarily used during development of the proposed system. It has been split into 16 movie excerpts for development, 16 movie excerpts for testing and the remaining 128 movie excerpts for training.

3.1.2 Hollywood Movie Corpus

The Hollywood movie corpus was introduced in [EWSS13] and had it's annotations refined in [LWS15]. It consists of four full-length Hollywood movies, ≈8 h in total. An overview of the movies is provided in table 3.1.
The original English versions of the movies were used, thus the speech is mostly in English. All of the movies feature a great deal of background noise as well as a rich soundtrack. The annotations discriminate speech and no speech and feature a precision of 10 ms. Notably the corresponding work was aimed at ASR, thus unintelligible voiced sounds like screams, groans etc. are annotated as no speech. By the definitions in [CRE+18] used in this work this qualifies as speech activity detection (SAD) rather than VAD.
Nevertheless, as it features annotations of actual movies this corpus is of great interest to this work. It has been used as unseen test data by [LWS15] and [HSN19]. Therefore it can be used to further evaluate the performance of the system proposed in this work.

3.2 Related corpora

The corpora in this section were not used in the training or evaluation of the proposed system. Yet, they generally are of interest to VAD and thus are listed for reference. While some are well promoted others were hard to find so this section might also be useful as a kind of knowledge base for corpora.

3.2.1 Music detection corpora

The corpora in this subsection are aimed at the detection of music, not voice activity. However, as this work is explicitely aimed at the audio tracks of movies, they are considered relevant enough to be outlined shortly.

Seyerlehner corpus

Lehner et al. describe a corpus of just below seven hours of broadcasts from Austrian television[SPS07]. The presence of music was manually annotated. The corresponding research is aimed at automatic music detection for royalty payments. Thus speech is not annotated and therefore the corpus is ultimately not useful for this work. However, because of its relatively small size and its careful annotation it was used during prototyping.

OpenBMAT

The Open Broadcast Media Audio from TV (OpenBMAT) corpus [MMG19] features more than 27 h of TV broadcast audio. Apparently unique, the loudness of music relative to other sounds is annotated. As speech is not annotated this corpus is mostly interesting to music detection research.

3.2.2 Record-wise labeled VAD corpora

As mentioned before, labels might be applied at frame, timespan or recording level. This subsection lists corpora wherein each recording belongs to exactly one class.

MUSAN

The MUSAN corpus [SCP15] features ≈109 h of recordings from different sources. Each recording is approximately 5 min long and belongs to either music (≈43 h, speech (≈60 h) or noise (≈6 h) class. In the corresponding work, a real time VAD system is tested by using small excerpts from the recordings. However, as this work is aimed at frame-wise labels, the MUSAN corpus was not used.

VAST

The VAST corpus is a very large multi-modal corpus created by the Linguistic Data Consortium. It was introduced in [TS18] and features ≈2444 h of amateur videos. Different parts of the source data were annotated with different purposes like language or speaker identification or ASR in mind. A total of ≈664 h featuring three languages was annotated for SAD. The annotations consider three classes: speech, no speech and music. However instead of frame-wise labels the data is split into segments each belonging to exactly one class. Parts of the source data containing music and speech are found in one segment belonging to the speech class and another segment belonging to the music class. Therefore the corpus is, although impressively large, unsuited for use in this work.

3.2.3 Artificial Corpora

As manually annotating media is a rather tedious and thus expensive approach, an easier way of acquiring training data is desirable. Corpora which where created using some kind of automated approach are listed in this subsection.

QUT-NOISE-TIMIT

One approach to artificially generate VAD corpora is to mix clean speech recordings with noise recordings, an approach pursued e.g. by [RMC07; EWSS13; DSVM10] with the latter exemplarily described here.
The QUT-NOISE-TIMIT corpus is rather large, consisting of ≈600 h of audio. It is mixed from the TIMIT corpus [GLF+92] released in 1992 by the US National Institute of Standards and Technology (NIST) and specifically recorded noise. The TIMIT corpus consists of ≈5 h of audio recordings of ten sentences each spoken by 630 speakers. The authors of the QUT-NOISE-TIMIT corpus note that there were no sufficiently large noise corpora available and they therefore recorded one. The noise consists of 20 scenarios and at least 30 min recording for each. These include public as well as private locations and a moving car. The TIMIT corpus and the noise recordings are mixed at different signal to noise ratios (SNRs), from −10 dB to 15 dB. Additionally for two scenarios the reverberant response was calculated and added to the speech as well.
For two reasons this corpus was not used for this work. Firstly, the sentences in the TIMIT corpus are ill-suited for the task at hand. They are calmly read as opposed to the emotionally charged speech often found in movies. And secondly, although mixing noise and clean speech provides noisy speech the result is quite different from movie audio. The noise was recorded in everyday environments, missing distinctive movie sounds like fighting. Music isn't mixed in at all. Therefore the corpus was not used in this work.

Large-Scale Multimodal Movie Corpus

Another way to obtain a corpus is by using a VAD system on a collection of media files. [YISK16] introduces a corpus of ≈2066 h from 1722 movies in 22 genres. They use MFCCs as features and a deep neural network (DNN) based classifier. The classifier is trained on ≈2.5 h of TV broadcasts. The training data was manually annotated. The VAD decisions are smoothed, including the dropping of voice segments shorter than 4 s. The resulting corpus is large but the annotations are very coarse and not reasonably precise. It is aimed at providing annotations where to look for dialogues in movies. Using this corpus as training data for VAD would presumably result in a system mimicking the less sophisticated VAD system used for its creation.

4 Related Work

This chapter provides a short overview of historic and recent approaches to VAD. With the available corpora covered in chapter 3, it focuses on features and classifiers. The fist section is a quick history of VAD, covering the time from the late sixties until the early 2000s. Afterwards recent approaches, sorted by input features, will be discussed. As mentioned before, MFCCs are the most common features for VAD, used by a lot of contemporary systems with some examples given in section 4.2. Preprocessing for VAD is being actively researched, therefore occasionally more exotic or specialized approaches are being published. Several of these can be found in section 4.3. Lately MRCGs have been used with promising results. The corresponding examples are discussed in section 4.4. An overview of all recent works covered here can be found in table 4.1.

4.1 Origins of voice activity detection

Voice activity detection has been a topic of interest for more than 50 years. At first the main application was the time-multiplexing of landline calls and data transmission. A US patent filed in 1967 already states "Statistical studies have shown that in a full duplex voice voice communication, each of the two channels remains idle, on the average, of 67 percent of time [...] this idle time is simply an exorbitant waste of the capabilities of the system"[FD67]. The patent further outlines a system to find these idle times and inject data transmissions. However, the actual detection of speech is only described as "A more detailed description of the decision circuit will be given later." Although no such description could be found, several other companies linked to telecommunications patented speech detection technologies in the mid-seventies. These systems, aimed for use in digital voice transmission, were purely electronic without software or programmable components. Some calculated frequencies based on zero crossing rate to determine the presence of fricatives[MP76; AO77]. Others compared the current signal amplitude to a threshold calculated from some kind of moving average presumed to be the noise level, considering any three to four frames above this threshold to contain speech[Jan77; LJZ77]. [AR76] already proposed five features to be used for voiced-unvoiced-silence classification: "the zero crossing rate, the speech energy, the correlation between adjacent speech samples, the first predictor coefficient from a 12-pole linear predictive coding (LPC) analysis, and the energy in the prediction error". In the following decade refinements and combinations of these approaches were developed[LU86]. Additional methods e.g. to avoid clipping were also introduced [Gru82]. Beginning in the late eighties periodicity measures, cepstral analysis and perceptually based linear prediction were

Table 4.1: Overview of related works in order of appearance.
If more than one system is described, the best is characterized in this table

Work	Input Features	Classifier	Distinctive Features
[TGY16]		LSTM	Comparison of different DNNs
[GG18]		BLSTM	Highly customized approach
[KSI18]	MFCCs	DNN	Multiple classes
[CLS+18]		FSTDNN	Sophisticated architecture
[HSN19]		FSTDNN	Best results on Hollywood corpus
[LWS15]	Handcrafted	SVM	Introduces Hollywood corpus
[JASJ17]	HPSS/MFCCs	DNN	Harmonic-percussive source sep.
[EWSS13]	RASTA-PLP	LSTM	Introduces LSTM for VAD
[CRE+18]	Spectrogram	FSTDNN	Introduces the AVA corpus
[ZSSP16]	Raw waveform	Complex NN	Raw waveform as input
[CWW14]		MLP	Introduces MRCGs
[ZW16]	MRCGs	Stacked DNN	Multi-resolution stacking
[KH18]		DNN	First attention mechanism for VAD
[LMHK20]		Complex NN	Two attention mechanisms

introduced to VAD [HM93]. Although proposed for ASR tasks in the late seventies [DM80], it wasn't until the early nineties MFCCs got picked up for VAD[ALB93]. Around this time the applications of VAD broadened. As mentioned, earlier systems had focused on the effective utilization of communication channels. Although noise reduction for ASR had been explored before, only now VAD was considered as an explicit component of ASR systems [HM93]. In 2003 an algorithm for retrieving MFCCs was standardized by the European Telecommunications Standards Institute [Eur03], further establishing them as commonly used features for VAD. To this day, although other features are used as well, MFCCs can be found in a lot of contemporary works on VAD [HSN19; GG18; Gra12; RLY13; KSI18].

While for the early systems the decision whether speech was detected or not was derived rather directly from some threshold, more sophisticated classifiers came into use during the eighties. The two-state-machines (speech - non-speech) were extended by adding a transitory state for more precise endpoint detection [Sav89]. To model the silence and talkspurt duration, the mean talkspurt rate and the overall speech activity, [Gru82] and [LU86] used probability density functions . Based on these probability density functions, the endpoint detection was refined leading to shorter talkspurts while still avoiding clipping. A generalized likelihood ratio test was incorporated into the decision rule by [SS98], improving the performance of VAD in scenarios with non-stationary noise. They also used Hidden Markov Models (HMMs) to calculate hangover. Further development led to [RSB+04] using a symmetric Kullback-Leibler divergence for their decision rule. By that time approaches utilizing machine learning for reference estimation were predominant, further manifested by the use of support vector machines e.g. by [RYGS06]. Even earlier [SSR01] had proposed a perceptron algorithm based classifier for VAD, foreshadowing the decades to come. And by 2004 [KRH04] proposed a VAD system featuring a MLP. Although using cepstral linear predictive coefficients instead of MFCCs, their system had fundamentally the same architecture as most of the latest approaches published.

As a result, about 15 years ago the general structure of today's VAD systems was in place. Since then progress was made in the area of even more sophisticated input features and mainly on the types and architectures of NNs used.

4.1.1 Origins of speech/music discrimination

Another field of research leading to the system in this work is speech/music classification (SMC). The goal of SMC is to classify recordings or single frames within recordings to be either music or speech[SS97; DC17]. Just like VAD, SMC can precede different higher-level tasks such as ASR, speaker recognition and song and musical genre recognition [DC17]. Related systems only try to detect music [SPS07] or annotate its loudness [MMG19]. These are aimed at digital rights management approaches to monitor broadcast media for copyrighted music. As for VAD, earlier systems focused on hand-crafted features. For example, [SS97] used eight different features based on energy, spectrum, cepstrum, zero crossing rate and beat detection. Recently, standard features like MFCCs combined with FSTDNNs are used [DC17].

4.2 MFCC-based

A well documented comparison of DNN, LSTM and FSTDNN classifiers can be found in [TGY16]. They used 24 log-Mel filterbank features with first order derivatives added were used as input features.A LSTM based classifier performed best at modeling context. It outperformed the other two considering equal error rate (EER) and AUC on both, seen and unseen noise conditions. So called "Noise-aware Training" was also applied. Therefore the average feature vectors of noise and speech frames were appended to the input vector. The preliminary classification was taken from a system without Noise-aware Training. This technique is shown to yield a significant relative improvement on all reported metrics.

A detailed description of a highly customized SAD system is found in [GG18]. A RNN called coordinated-gate LSTM (CG-LSTM) by the authors was used as a classifier. As described in section 2.2.4, the activation of each LSTM gate is based on the input, and the cell state and output of the previous step. The CG-LSTM architecture uses so called "peephole vectors" to calculate the activation of the gates. Thus, the activation of each gate is additionally based directly on the previous activation of all the gates. This system was fed with MFCCs. The parameters of the feature extraction however were learned instead of being set manually. Namely the window type and size, minimum and maximum frequency, number of filters, number of discrete cosine transform (DCT) coefficients and the context width for Δ- and $\Delta\Delta$-features. To achieve this, the quantum-behaved particle swarm optimization algorithm, based on the popular particle swarm optimization algorithm was used. This algorithm can thus be used to effectively learn the parameters of non-differentiable processes. The parameters of the CG-LSTM were also learned using the quantum-behaved particle swarm optimization, but subsequently improved using the SMORMS3 algorithm. SMORMS3 was found to be superior to Adam. To optimize the SAD system towards ASR, a specialized loss function was used.

It is based on the substitutions, deletions and insertions in the output of a subsequent ASR system with respect to the ground truth. And finally the 100 utterances with the highest error rate were added to each minibatch to emphasize them during training. An adaption of this technique was used in this work.

Normally, VAD is a binary classification problem. However, it might be beneficial to discriminate between different background noise conditions. This was done by [KSI18]. They used a DNN based classifier on top of 13-dimensional MFCCs with Δ- and $\Delta\Delta$-features. The target classes were "speech", "music", "singing", "noise" and "silence". The authors report performance gains by the increase in classes as opposed to binary classifiers or their own experiments with three or four classes.

Another approach focusing on the architecture of the NN based classifier is reported in [CLS+18]. It is based on 40-dimensional log-scale MFCCs and a FSTDNN as base architecture. So called "causal dilated convolution" was used. The "causal" simply implies, that only the left context was used. This decision had been made, because the work was aimed at endpoint detection in Google home devices and Android voice typing, which is a latency-dependent task. The "dilated" describes the filter kernel skipping some frames, leading to a broader context while maintaining a reasonably trainable input size. Gated activations, similar to LSTMs, are also used. Therefore, two sets of weights, one for the activation, the other for the filters are learned. Finally, residual connections skipping entire layers were used to handle the vanishing gradient problem, allowing for up to 36 hidden layers to be trained.

The system proposed in this work was trained on the AVA corpus and evaluated on the Hollywood corpus. Both were used for testing in [HSN19]. They report on several complex architectures. One model was a single BLSTM layer with a DNN with successively smaller layers on top. Four additional models based on FSTDNN stacks and differing DNN based networks on top are also presented. One of these, with so called "time distributed" fully connected layers on top of the FSTDNN stack is reported to perform best, but only by a small margin. The results are compared to the systems proposed in this work in chapter 7.

4.3 Less common input features

As mentioned before, although MFCCs became the most common input features for VAD, specialized input features are still actively being researched. Some recent approaches will be discussed in this section.

The fine-grained annotations for the Hollywood dataset were first mentioned in [LWS15]. The authors also included an evaluation of their own system. Its classifier is a support vector machine as implemented by the Weka toolkit [HFH+09]. To achieve competitive results, they utilized an elaborate feature set. It was composed of the fluctogram, the spectral flatness, the spectral contradiction and the so called "polynomial shape spectral contrast" of the signal. This resulted in an overall 126-dimensional feature vector. However, the system proposed in this work outperformed their system, using less complicated features but a more powerful classifier.

A pair of intuitive characteristics of audio signals are their harmonic and percussive components. An attempt to use these for VAD is described in [JASJ17]. They use

a technique described in [Fit10] to first separate the input into its percussive and harmonic components. This is achieved by median filtering along either the time or the frequency axis of a spectrogram. From the filtered spectrograms 13-dimensional MFCCs are extracted. The classification is done by a DNN based classifier. However, the reported results on the Hollywood corpus are inferior to [LWS15] and [HSN19].

Although LSTMs have been around since 1997 [HS97], they were not used for VAD until 2013. The first attempt to do so is described in [EWSS13]. As input they use relative spectral transform perceptual linear prediction features with first order derivatives appended. Two classifier networks are described: One with a single recurrent layer with 200 LSTM cells. The other with three layers, a LSTM layer with 50 cells, a sigmoid layer with ten units and a final LSTM layer with 20 cells. From a 2020 perspective, these are relatively small networks. However, a performance superior to other contemporary approaches is reported. The movies of the Hollywood corpus were first used in [EWSS13] as well, but the supplied annotations are very coarse. Therefore the annotations by [LWS15] were used by all subsequent works found.

The systems proposed in this work were all trained using the AVA corpus. In the work introducing the AVA corpus [CRE+18], two FSTDNN based VAD systems were also described. The one yielding better results is based on the ResNet-50 architecture [HZRS15] introduced for image recognition. It features 50 layers of FSTDNNs with residual connections to handle the vanishing gradient problem and keep the system trainable. As input a Mel scaled spectrogram was used.

With FSTDNNs becoming increasingly popular for VAD, spectrograms are often used as input without further preprocessing. One step further is the approach by [ZSSP16], applying no preprocessing at all. The input to the classifier is the raw waveform. It is first fed into a TDNN and subsequently filters along the frequency axis are applied. A small LSTM with two layers of 48 cells and a single feedforward layer with 48 units then lead to a VAD decision. The authors report relative improvements of more than 50 % over a LSTM-only model for noisy data.

4.4 MRCG-based

A comparative study of no less than 16 different input features is described in [CWW14]. MRCGs are introduced as a novel feature. They are aimed at incorporating the context required for a robust VAD decision directly into the input for each frame. A MLP is used as classifier. The authors report superior performance to all other tested features, including a combination of MFCCs, relative spectral transform perceptual linear prediction and the amplitude modulation spectrogram.

The incorporation of contextual information is a key component of robust VAD. A complex, yet interesting approach is described in [ZW16]. The proposed system is a DNN based classifier with MRCGs as input features. It exhibits two distinctive characteristics. The first is the use of so called “boosted DNNs”. The idea is to produce multiple predictions from a single DNN by “boosting” the contextual information. This is achieved by training the network to not predict a single label on a single frame, but a sequence of labels on a sequence of frames. The final prediction

is the average of the predictions for a frame from different time steps. A similar technique was used by [HSN19], but the rise in computational cost was considered disproportionate compared to the performance gain. The second interesting technique used by [ZW16] is the so called "multi-resolution stacking." A stack of DNNs was used as a classifier. Each block received the output of the previous block and a different context window of frames as input. The context got increasingly wider. To limit the size of the input vector, intermediate frames were skipped. The input vector in each time step is constructed from the transposed audio features by taking equally sized steps on both side of the current frame and its direct neighbors. The authors [ZW16] report a significantly increased performance compared to prior works. They also emphasize the good generalization capability towards noise unseen during training.

Attention mechanisms have successfully been applied to varying fields, including machine translation and ASR [CBS+15]. The first attempt at using them for VAD was described by [KH18]. Generally speaking, they use a MRCGs-based DNN classifier. Their approach is based on the previously described work [ZW16], using the same spacing of feature vectors. They also generate their final prediction as the average of the predictions for a single frame from several sequence to sequence predictions. However, they replaced the complex and computationally expensive multi resolution stacking with an attention mechanism. They also used specialized loss functions to focus the attention on the most crucial frames, particularly in high noise conditions. They report performance on par or slightly superior to [ZW16].

Another work, based on both [ZW16] and [KH18] is [LMHK20]. They used the same spacing of feature vectors as the other two, but did not transpose them. Instead, they used several FSTDNNs, deployed in pairs, connected similarly to [CLS+18]. They called this approach "spectral attention". Subsequently, multi-head attention [VSP+17] is applied, called "temporal attention" by the authors. They reported significantly improved performance in comparison with [ZW16] and [KH18].

5 Proposed Systems

This chapter describes two structurally different systems, that performed well on the evaluation data, compared to related works. The experiments leading to these exact configurations are described in section 6.1. In the following section the structure of the proposed systems will be described. In the second section the used software and other implementation details are covered.

5.1 Classifier Networks

The VAD classification was performed by different NNs. Numerous configurations were tested and the two best compared to other works in section 6.2. This section will outline, which parameter combinations were tested, and which proved to perform best. The best performing models are depicted in fig. 5.1. For detailed results on the experiments conducted see section 6.1.

5.1.1 BLSTM-based

The first proposed system consists of a multi-layer BLSTM and feedforward layers for dimensionality reduction. Several different input features were tested, with 40-dimensional MFCCs yielding the best performance.

Models with differing numbers of layers, one to eight, were tested. Likewise, the number of units per layer was varied between 256, 512, and 1024. An attempt was made to test a system with 2048 hidden units, but the available RAM of the GPU was exceeded. The best AUC values were achieved by a model with four BLSTM layers of 1024 hidden units each. Models with four layers, but 512 and 256 hidden units each, scored second and third. Deeper models achieved a lower AUC, but were still superior to models with less than four layers, indifferent of the number of hidden units per layer.

Three ways of combining the outputs of the two directions of the BLSTM were tested. Following [MSW18] a pairwise maximum was computed. However, the results proved inferior to simply appending the directions into a single vector. Ultimately, summing the two directions showed the best performance. The resulting vector was fed to a single feedforward layer to be projected onto four outputs. A softmax layer transformed these into a probability distribution of the possible output classes.

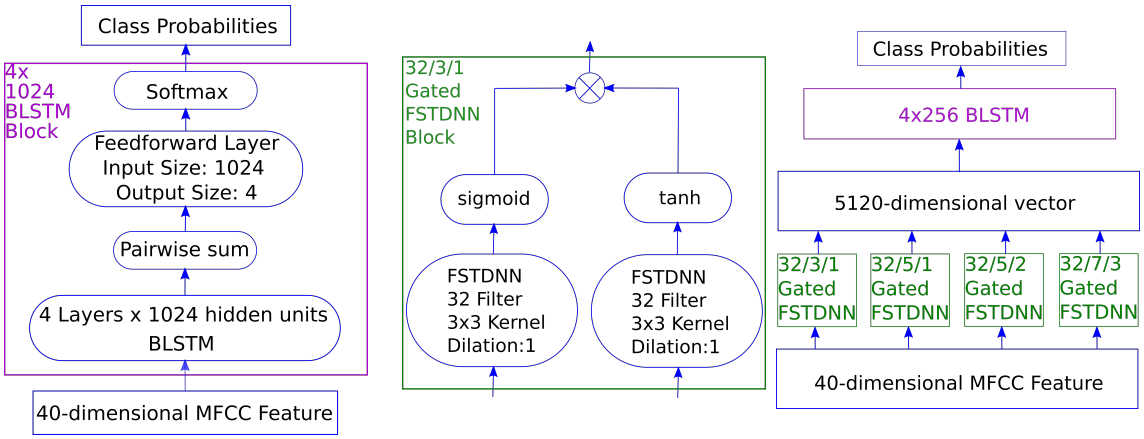

Figure 5.1: Structure of proposed models

To the left, the structure of the proposed BLSTM-based system. The structure of the violet part, although with different parameters, is reused in the proposed FSTDNN-based system. In the middle, a basic building block of the proposed FSTDNN-based system, called "gated FSTDNN". To the right, the structure of the proposed FSTDNN-based systems. Note that the four gated blocks all have different kernel sizes and dilation.

The best performing variant of a pure BLSTM-based model is depicted on the left side of fig. 5.1. In the remaining chapters, this model will be referred to as "BLSTM-based" or simply "BLSTM".

5.1.2 FSTDNN-based

The second proposed system is based on several FSTDNN units. Again, several different input features were tested. For this system, 64-dimensional MFCCs yielded the best performance.

The model itself is similar to the model proposed by [CLS+18]. There, the technique described below was called "gating". This work uses "gating" and "gated" in the same way. Note that [LMHK20] employed a very similar technique, but called it "temporal attention".

The gated FSTDNN blocks are depicted in the middle of fig. 5.1. Each one consists of two identical FSTDNNs. Both are fed with the same input. A sigmoid function, and a hyperbolic tangent function (eq. (2.1)) respectively, are applied to the outputs of the FSTDNNs. The output of the gated FSTDNN block is an element-wise or Hadamard product [GBC16, p. 34] of the outputs of the two nonlinearities.

Another method taken from [CLS+18] are the increasing filter sizes and dilations. As depicted, the leftmost gated FSTDNN block has a filter size of 3x3 and no dilation. The rightmost block has a filter size of 7x7 and a dilation of 3, spanning a height and width of 21x21 values. Thus, increasingly broader, yet coarser context information can be incorporated. As opposed to [CLS+18], the gated FSTDNN blocks are not stacked with residual connections. Instead, their results are concatenated and fed to a smaller version of the BLSTM-based model.

Table 5.1: EERs and corresponding thresholds for proposed systems. These values were calculated on the AVA validation set and used for testing on the Hollywood corpus.

	EER	Threshold
BLSTM-based	0.095	0.566
FSTDNN-based	0.092	0.47

Several combinations for the number of filters in each gated FSTDNN block, as well as for the size of the subsequent BLSTM block were tested. The detailed results are listed in section 6.1.3. The right side of fig. 5.1 shows the best performing configuration. In the remaining chapters this model will be referred to as "FSTDNN"-based or simply "FSTDNN".

5.2 Postprocessing

The smoothing techniques outlined in section 2.4 were all tested. Details on parameters and results can be found in section 6.1.4. Ultimately, smoothing provided no benefit with respect to the chosen metrics. Thus, it was not employed in the proposed systems. Note however, that smoothing might prove beneficial, if one of the systems was used as a preprocessor for ASR or a similar task.

Thresholding as described in section 2.4 was applied with the EER as target. The corresponding threshold are listed in table 5.1

5.3 Implementation

This section describes, how the proposed system was implemented, which software and libraries where used and how certain parameters were chosen. If parameters were fixed early on, they are reported here. If sufficient experiments were performed, they are listed in chapter 6.

The proposed systems were entirely implemented in python 3.8 using version 1.6 of the pytorch library [PGM+19]. Within the pytorch library, most of the functionalities needed for NN based systems were included. Pytorch is aimed at being as compatible as possible with numpy. Thus, if a certain functionality was unavailable in pytorch, it was almost always possible to use numpy functions without further ado.

Pytorch is capable of using the CUDA API [NVF20] to utilize GPUs for NN computation. Depending on which machine the code was run, version 9.2, 10.1 or 10.2 of CUDA were used. As far as the implemented systems were concerned, no difference was encountered. The management of the corresponding versions of pytorch, and other python components, was handled by the Anaconda virtual environment management system [20].

5.3.1 Preprocessing

The MFCCs were extracted using the corresponding class of the torchaudio package, which is part of pytorch. The number of frequency bins was left at the default value of 400, except when extraordinary long windows were tested. Then the number of frequency bins was set to the number of samples in the given excerpt of the audio stream. If not specified otherwise, the window length was set to 25 ms, and the window shift to 10 ms. The default pytorch MFCCs use a dB-scaling before the DCT is applied. For this work, standard log-scaling as described in section 2.1.1 was used instead.

For the generation of MRCGs, the pycochleagram implementation by [mcd20] was used. The frequency filters were limited to stay below 8 kHz. This was necessary, as the audio files were resampled to 16 kHz. A decibel scale was used as nonlinearity during cochleagram creation. The included polyphase resampling proved to be very resource hungry, pushing the time needed for preprocessing an audio file to almost 20 times its duration. To find a more efficient solution, the cochleagrams of several frames without resampling were analyzed. There was little to no change along the time axis. Thus a simple mean-averaging was used to generate 64-dimensional feature vectors from the original cochleagrams. The MRCGs were then generated as described in section 2.1.2.

5.3.2 Training

Adam [KB17] and SMORMS3 [Sim15] were used for training. The Adam implementation provided by pytorch was slightly adapted to return the step size. For SMORMS3 the implementation by [19] was used and also slightly adapted.

6 Experiments

The proposed systems were trained using 128 of the 160 movie excerpts of the AVA corpus. Another 16 excerpts were used as validation set. The remaining 16 excerpts were used as unseen test data to estimate system performance. The validation- and testset were drawn at random once, but remained the same for all training and test runs. As no work using only part of the AVA corpus for testing is known, these results are only used for comparison of different proposed systems. Notably, this data is used to evaluate how the different classes affect the system performance. For comparison with related works the Hollywood corpus is used. This corpus was also used as a testset by several other works [HSN19; JASJ17; LWS15].

Throughout this section, the best performing proposed models are used to present the effects of the variation of certain parameters. These models have been described in section 5.1 and are denoted "BLSTM-based" and "FSTDNN-based".

The remainder of this chapter will first describe the experiments that led to these two configurations. Subsequently, they are evaluated against related works on the Hollywood corpus. The last section will give some insights, on how the models deal with noise and background music respectively.

6.1 Development of proposed systems

This section describes the effects of the variation of certain parameters of the proposed models. The results are usually provided in terms of AUC as metric. AUC was chosen as most of the experiments were conducted without setting a threshold. Thus, the F1-Score is a poor metric of the performance, as it can most easily be tuned. The parameters not explicitly mentioned are those listed in section 5.1.

6.1.1 Preprocessing

All recordings were resampled to 16 bit, 16 kHz mono .wav files. MFCCs, cochleagrams and MRCGs were extracted once and stored for later use. MFCCs of different dimensionalities, 32, 40, and 64, were tested. Inspired by the MRCGs, a feature denoted "MRMFCC" was tested. It consists of two sets of MFCCs as input. One was derived from a standard window of 25 ms. For the other, a window ten times longer was used. The shift between frames was kept at 10 ms, as to produce an identical number of frames. The results are listed in table 6.1. As can be seen, 40-dimensional MFCCs proved optimal for the BLSTM-based classifier,

Table 6.1: AUCs on the AVA test set for different input features

Feature	MFCC			CG	MRCG	MRMFCC	
Dimensions	32	40	64	64	4x64	2x40	2x64
BLSTM-based	0.958	**0.965**	0.954	0.924	0.369	0.94	0.94
FSTDNN-based	0.964	0.967	**0.968**	0.967	0.509	0.954	0.958

while 64-dimensional MFCCs were best for the FSTDNN-based classifier. Thus, these combinations of input features and classifiers were chosen for all subsequent experiments.

The MRCGs yielded mediocre results. There are two possible explanations: The implementation of the MRCGs extraction might be erroneous. There was only research level code available for the extraction of cochleagrams, let alone MRCGs. Thus, although great care was taken when trying to extract them, mistakes might have happened. As the extraction of MRCGs for the training corpus took several days, it was impossible to test the effect of some parameters thoroughly.

However, the results listed in table 6.1 also allow for a different explanation. As can be seen, the FSTDNN-based systems performed disproportionally better with MRCGs than the BLSTM-based. This might imply, that only the combination of MRCGs with appropriate FSTDNNs yields the promising results reported in [ZW16], [KH18] and [LMHK20].

6.1.2 Training

Each proposed system was trained for up to 50 epochs. Early stopping was used if the average improvement of the CEL on the validation data fell below 0.02 for five epochs, starting from the fifth epoch. The actual change, not the absolute, was used, thus the early stopping triggered relatively fast when the CEL worsened. The best model based on FER on the validation data was saved and used for subsequent testing. This was not necessarily the model after the last epoch. Following [GG18], the ten minibatches yielding the worst CEL during an epoch were presented a second time. Both Adam [KB17] and SMORMS3 [Sim15] were tested as schedulers. The latter consistently yielded faster convergence. All results listed here were achieved by systems trained using SMORMS3.

For training, as well as for evaluation, the recordings were split into samples with a duration of 10 s, i.e. 1000 frames. During training, the samples were provided in randomized minibatches of 20 samples each. During evaluation, all samples belonging to a single recording were presented sequentially.

6.1.3 Network configuration

To determine the best network configuration, extensive testing of different parameters was conducted. For several of the tested configurations, the AUC on the AVA

Table 6.2: AUCs for selected configurations of the BLSTM-based system.

BLSTM-Layers	2	4			6		8	
Hidden Units	1024	256	512	1024	256	512	256	512
AUC	0.961	0.963	0.964	**0.965**	0.957	0.962	0.961	0.962

test set will be listed. The AUC was chosen as metric for two reasons. First, as stated before, it is independent of the easily adjustable threshold, as opposed to the F1-Score. Second, several other metrics showed generally similar orderings of the tested systems, but a higher variance. Thus, the AUC was considered a relatively robust metric of system performance.

The results presented here are all based on the combination of classifiers and MFCCs described in section 6.1.1. Several of the presented configurations have also been tested with different input features. However, the results reinforced the decision to use 40-dimensional MFCCs for with BLSTM-based models and 64-dimensional MFCCs with FSTDNN-based models for further experiments. The results are presented for BLSTM-based and FSTDNN-based systems separately.

BLSTM-based

For the BLSTM-based systems two parameters of the model itself could be optimized: The number of layers, or depth, of the network. And the number of hidden layers, or width, of the network. A selection of the tested configurations is listed in table 6.2. The best configurations all feature four layers. Within this group, models with more hidden units performed better. An attempt at testing a model with one layer and 2048 hidden units was made, but failed because of insufficient RAM. Thus, four layers of 1024 hidden units each was chosen as the final configuration. The next decision was how to combine the outputs of the two directions of the BLSTM. Three ideas were tested. First, the outputs were simply concatenated. The following layer then was a feedforward layer projecting from two times the hidden size of the BLSTM onto four outputs. The second idea was to sum the forward and backward values for each position. The following layer then had an input size equal to the hidden size of the BLSTM. And third, following [MSW18], taking the pairwise maximum of the two outputs. These possibilities were tested and the results are listed in table 6.3. The pairwise maximum yielded the worst results. Concatenating the outputs or summing them yielded comparable results. Summing was chosen as the method to be used in the final model. It showed the best performance while at the same time being fastest.

The already mentioned feedforward layer and a softmax function constitute the output of the classifier. A smaller version with only 256 hidden units per unit was reused as part of the FSTDNN-based layer described next.

Table 6.3: AUCs for different methods of combining BLSTM directions.

Hidden Units	512			1024		
Combination	Concatenate	Sum	Max	Concatenate	Sum	Max
AUC	0.964	**0.965**	0.962	**0.965**	**0.965**	0.962

Table 6.4: AUCs for selected configurations of the FSTDNN-based system The AUCs were calculated on the test data from the AVA corpus.

Filters per FSTDNN	8	16	32				48
BLSTM layers	4	4	2		4		4
BLSTM hidden units	256	256	256	512	256	512	256
AUC	0.964	0.965	0.966	0.964	**0.967**	**0.967**	0.965

FSTDNN-based

Before the decision towards the architecture described in section 5.1.2 was made, several other models were tested. This included a stack of non-dilated FSTDNNs, a stack of FSTDNNs with residual connections and models based on TDNNs. However, none was able to outperform the previously described BLSTM-based system. If this was due to badly chosen parameters or the general architectures considered remains an open question. After the first promising results using the proposed FSTDNN-based architecture, research was focused on the optimal parameter configuration rather than other architectures.

On top of the proposed gated FSTDNN blocks, a smaller variant of the BLSTM-based system was used. The actual size was determined through testing again. The only parameter of the FSTDNNs that was varied was the number of filters. In table 6.4 the tested parameter combinations are listed. The number of filters yielding the best results was found to be 32. Again, four layers of BLSTM performed best. The number of hidden units did not yield mayor differences, thus the computationally less expensive 256 was chosen. The remainder of the BLSTM block was not altered, utilizing the same composition of sum, feedforward layer and softmax function to generate class probabilities.

6.1.4 Postprocessing

All smoothing techniques described in section 2.4 were tested with both systems. The decision, not to include smoothing in the proposed systems, was made based on the results on the AVA validation data. However, the results reported here were calculated on the Hollywood corpus. Thus they are better comparable to the results reported in table 6.6 and table 6.7.

Table 6.5: Effects of differing smoothing techniques
These F1-Scores were calculated on the Hollywood corpus. The classifier was the BLSTM-based classifier described in section 5.1.1

Smoothing	None	Continuous	Continuous + Discrete	Discrete	Maxvote
BLSTM	**0.77**	0.769	0.755	0.754	0.556
FSTDNN	**0.777**	**0.777**	0.761	0.761	0.542

The effects of the different smoothing techniques on the Hollywood corpus in terms of F1-Score are listed in table 6.5. The effects of maxvote smoothing were disastrous. Discrete smoothing was better, yet still had a measurable negative impact on F1-Scores. The effects of continuous smoothing were barely noticeable. Thus, the proposed systems do not apply any smoothing. However, these results are based on the F1-Score. A subsequent ASR system might still benefit from continuous or maybe even discrete smoothing.

6.1.5 Comparison of ROC curves

The ROC curves of both systems are shown in fig. 6.1. The right diagram is an enlarged version of the upper left corner of the left one for better comparison. As can be seen the two systems behave very similar. With coordinates given as (FPR, TPR) the curves are roughly identical between (0, 0) and approximately (0.02, 0.65). There is also little difference between approximately (0.31, 0.93) and (1, 1). These intermediate points represent reasonably good performances themselves. The biggest difference is at an FPR of approximately 0.15, with an TPR of approximately 0.85 for the BLSTM-based system and 0.87 for the FSTDNN-based system. However, the threshold of each system needs to be chosen with great care to exploit this improved performance.

6.2 Evaluation on Hollywood corpus

No other work before used part of the AVA corpus for training and another part for testing. Thus, another corpus was needed for evaluation. The Hollywood corpus, a corpus consisting of four Hollywood movies annotated by [LWS15] was used. In table 6.6 the results on this corpus are coarsely compared to other works. Another, more detailed comparison can be found in table 6.7. However, these detailed values were not reported by [HSN19], containing the most recent and performant systems. In-depth analysis of both tables constitute the remainder of this section.

6.2.1 Coarse comparison

One work, [HSN19], tested several systems on this corpus. The two best were selected for comparison. Both use the vgg-conv network, originally developed for

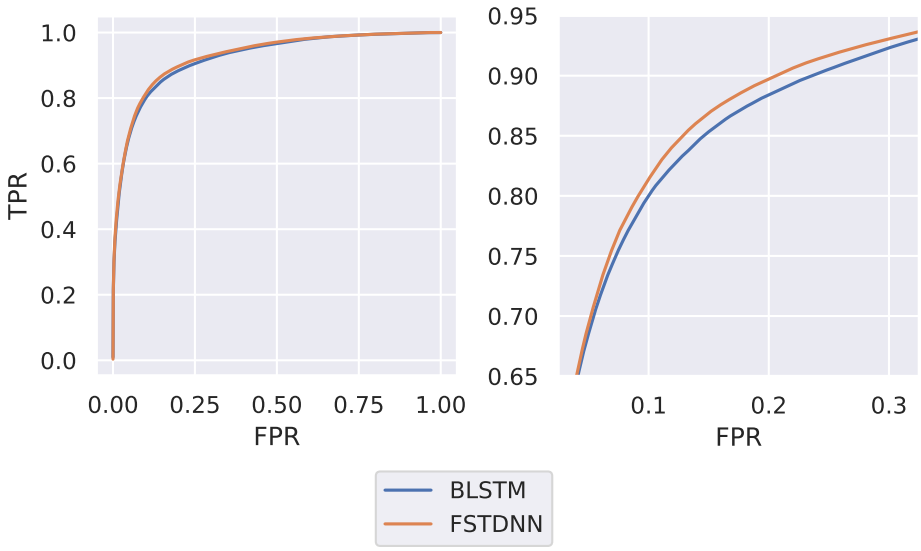

Figure 6.1: ROC curves of the proposed systems.
The right diagram shows a detailed view of the upper left corner of the left diagram.

image recognition [SZ15]. The vgg-conv network employs several layers of FSTDNNs with pooling layers in between. One of the systems described in [HSN19] features a single BLSTM layer, another pooling layer and two feedforward layers on top of the vgg-conv block. Thus, it is very similar to the FSTDNN-based network proposed in this work, yet more complicated. In table 6.6 it is denoted as [HSN19](1). The other system, denoted [HSN19](2), features only feedforward and pooling layers on top of the vgg-conv block.

The other two works listed in the table mainly rely on sophisticated input features. A decomposition of the input audio into harmonic and percussive components is used by [JASJ17]. As classifier they employ a simple DNN, consisting of three layers with 286 hidden units per layer. The classifier used by [LWS15] is a support vector machine. They employ a highly customized 126-dimensional input vector.

The first system listed in the table, denoted "WebRTC" is the voice activity detector of the widely used WebRTC framework [Goo20] This framework is aimed at in-browser real-time communication. Therefore it is not at all optimized towards VAD in media audio. However, it is a highly optimized, widely used VAD system. It features a thresholding parameter, called "aggressiveness", allowing values between zero and three. The higher the value, the less frames are classified as containing voice activity. The aggressiveness was set to three for this comparison, thus the relatively low recall. However, for lower aggressiveness values the precision deteriorated so badly the F1-Score got even worse. Thus, the system is not competitive, but stands to prove one thing: that VAD on media audio is hard.

Table 6.6: Results on the Hollywood corpus

	Accuracy	Precision	Recall	F1-Score
WebRTC	0.37	0.49	0.59	0.54
[HSN19](1)	**0.89**	0.79	0.75	0.77
[HSN19](2)	0.88	0.74	**0.81**	0.77
[JASJ17]	0.86	**0.84**	0.57	0.67
[LWS15]	0.87	0.75	0.73	0.74
BLSTM	0.88	0.75	0.79	0.77
FSTDNN(1)	0.88	**0.84**	0.66	0.73
FSTDNN(2)	**0.89**	0.76	0.8	**0.78**

The systems proposed in this work are denoted as BLSTM, FSTDNN(1) and FSTDNN(2). The FSTDNN(2) is the same system as FSTDNN(1), but with a threshold tuned slightly in favor of the Hollywood corpus. This was done because the threshold calculated on the AVA validation data generalized well for the BLSTM system, but not for the FSTDNN system. Thus, a system showing a very good performance in terms of AUC and accuracy had a mediocre recall and thus a unfavorable F1-Score as well. However, this is an only partly fair comparison and with respect to F1-Score the BLSTM-based system should be considered superior.

The best accuracy was achieved by both the first system from [HSN19] and the specially trimmed FSTDNN-based system. The second system by [HSN19], the untrimmed FSTDNN-based system and the BLSTM-based system make a close second, with the two remaining specialized systems only slightly worse. Thus the accuracy shows little variance and is not very useful for a ranking of the systems.

The precision and recall need to be examined together. A recall of 0 automatically yields a precision of 1. A system is considered good, if it can raise the recall without the precision deteriorating or vice versa. Thus, the high precision of [JASJ17] is not as impressive as it may seem at first. They simply classified fewer frames as containing speech as others. This holds true for the untrimmed FSTDNN-based system as well. Impressive results are achieved by [HSN19](1), yielding the third highest recall while maintaining a competitive recall. Also by [HSN19](2), yielding the highest recall and still a reasonable precision. And by the BLSTM-based system proposed in this work, yielding the third highest recall while maintaining a competitive precision. The trimmed FSTDNN-based system is also very good, but as mentioned before this is not a fair comparison.

The F1-Score is the most informative value in this table. As it is calculated from those, it exactly resembles the ranking to be expected from precision and recall. The specially trimmed system is best, the two systems by [HSN19] and the BLSTM-based system share the second place. It should however be noted, that the BLSTM-based system is much less complicated than the systems by [HSN19].

6.2.2 Film-wise comparison

Detailed results for several metrics and all four movies in the Hollywood corpus can be found in table 6.7. For a short overview of the movies see table 3.1. The systems featured in the table are the untrimmed FSTDNN-based system and the BLSTM-based system. Further the systems by [HSN19] and [JASJ17] as described in section 6.2.1. Such detailed results were not reported by [HSN19], thus their systems are omitted.

Both the highest precision and lowest false positive rate (FPR) are mostly achieved by [JASJ17]. However, as before the corresponding recalls are low and the false negative rates (FNRs) high. Their system simply classified few frames to contain voice activity. Therefore, the F1-Scores they achieved are not competitive. As they did not report AUCs, those can not be compared.

The best AUCs were consistently achieved by the FSTDNN-based system. The second best values, and for "Kill Bill" even the same was achieved by the BLSTM-based system. Somewhat surprisingly, the best EER for both "I am Legend" and the whole corpus was achieved by [LWS15]. Their overall performance on "I am legend" is impressive. Their system also did very well on "Kill Bill", a trend seen in both of the proposed systems as well. Notably, these are the movies with the lowest percentage of speech. Thus, all the systems appear to profit from class imbalance with less speech frames, but the system by [LWS15] more than the two proposed systems.

The BLSTM-based system consistently yields the highest recalls and F1-Scores. The FSTDNN-based system yields high precision, but at the cost of significantly lower F1-scores. The system by [HSN19] keeps a good balance of precision and recall but is generally inferior to the BLSTM-based system.

6.3 Effects of noise and music

This work is especially aimed at VAD in the presence of music. As reported before, no other work using parts of the AVA corpus as train- and testset respectively could be found. Thus, a comparison of the performance with respect to different classes can only be drawn between different tested systems. The results of such a comparison are depicted in fig. 6.2. Note that the scale for the TPR only covers the range from 0.9 to 1. All these values are very close together. The depiction overstates the differences for better comparability. The four groups of bars belong to one system each as denoted below the x-axis. Each color represents one class, see the legend in the lower right for details.

The chosen metric was the TPR for a fixed FPR of 0.315. This means a threshold was chosen such, that 31.5 % of the frames not containing any voice activity would still be placed in one of the classes meant to contain voice activity. Subsequently for all the frames belonging to one of the classes meant to contain voice activity the rate of them being classified as containing voice activity was calculated. Whether they would have been assigned to the actually correct class was not questioned, only if it contained speech or not.

Table 6.7: Metrics per movie

The Bourne Identity								
Model	AUC	EER	FNR	FPR	ACC	PREC	REC	F1
[LWS15]	0.897	0.134	0.297	0.074	0.866	0.776	0.703	0.738
[JASJ17]	-	-	0.355	**0.04**	0.876	**0.855**	0.645	0.736
BLSTM	0.92	0.148	**0.194**	0.091	0.881	0.763	**0.806**	**0.784**
FSTDNN	**0.928**	**0.133**	0.282	0.049	**0.889**	0.841	0.78	0.775
I am Legend								
Model	AUC	EER	FNR	FPR	ACC	PREC	REC	F1
[LWS15]	0.922	**0.092**	0.225	0.062	**0.908**	0.738	0.775	0.756
[JASJ17]	-	-	0.463	**0.023**	0.897	**0.842**	0.537	0.656
BLSTM	0.949	0.117	**0.178**	0.075	0.906	0.712	**0.822**	**0.763**
FSTDNN	**0.95**	0.107	0.397	0.027	0.905	0.834	0.603	0.7
Kill Bill: Volume 1								
Model	AUC	EER	FNR	FPR	ACC	PREC	REC	F1
[LWS15]	0.914	0.124	0.186	0.109	0.877	0.640	0.814	0.717
[JASJ17]	-	-	0.313	0.072	0.882	0.695	0.688	0.691
BLSTM	**0.944**	**0.116**	**0.163**	0.082	**0.902**	0.713	**0.837**	**0.77**
FSTDNN	**0.944**	0.124	0.328	**0.048**	0.897	**0.772**	0.472	0.719
Saving Private Ryan								
Model	AUC	EER	FNR	FPR	ACC	PREC	REC	F1
[LWS15]	0.880	0.155	0.306	0.084	0.845	0.797	0.695	0.742
[JASJ17]	-	-	0.535	**0.02**	0.814	**0.916**	0.465	0.617
BLSTM	0.912	0.167	**0.26**	0.087	**0.858**	0.802	**0.74**	**0.77**
FSTDNN	**0.923**	**0.151**	0.374	0.049	0.845	0.861	0.626	0.725
All								
Model	AUC	EER	FNR	FPR	ACC	PREC	REC	F1
[LWS15]	0.895	**0.13**	0.271	0.082	0.870	0.748	0.729	0.738
[JASJ17]	-	-	0.43	**0.037**	0.861	**0.838**	0.572	0.67
BLSTM	0.924	0.148	**0.214**	0.087	**0.881**	0.753	**0.786**	**0.769**
FSTDNN	**0.929**	0.14	0.345	0.043	**0.881**	0.835	0.655	0.734

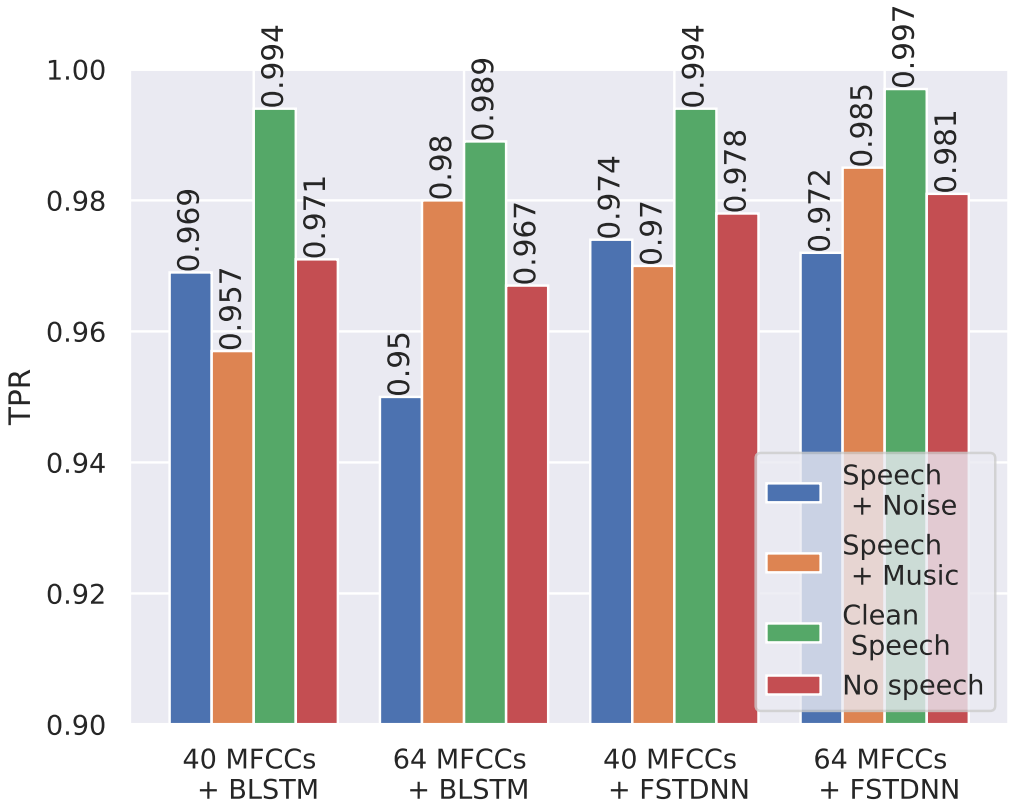

Figure 6.2: True positive rates for different classes.
The true positive rates are reported for a threshold at which the same class would achieve a false positive rate of 0.315.

One obvious result is, that the FSTDNN for every class and combination with one exception for speech with music. However, as the AUC of those systems was already known to be higher this is no surprise.

More interestingly, the differences between the classes for speech with noise and speech with music are less obvious for the FSTDNN-based systems. It is possible that the FSTDNNs were better capable to learn characteristics of voice activity independent of the type of background noise than the BLSTM-only system.

The most interesting find however is, that 64-dimensional MFCCs are superior for distinguishing speech and music. 40-dimensional MFCCs on the other hand perform better at distinguishing speech and noise. This is backed by additional data from training runs not reported here to maintain conciseness. As both features were extracted from the same number of frequency bins, higher dimensional MFCCs meant a better spectral resolution at the cost of temporal information. Thus the results might imply, that music can be distinguished from speech robustly based on fine-grained spectral analysis while noise requires more temporal information. However, further research is required to achieve a reliable confirmation of this assumption.

7 Discussion

This work is aimed at voice activity detection in media audio, especially in the presence of background music. To achieve this, two systems have been proposed. Both consist of a preprocessing step yielding input features and a NN-based classifier. In chapter 6 the decision to use the reported architectures and parameter combinations was justified. Furthermore, the performance of the proposed systems was compared to other contemporary approaches. In this chapter, the overall performance will be summed up and put into perspective. Moreover, possible implications of the results and possibilities to extend the proposed systems will be examined.

The overall performance of the proposed systems is competitive. [HSN19] reported comparable results, but used a larger and more complicated NN architecture. The results reported by [JASJ17] and [LWS15], although impressive, are not competitive.

Several input features were tested in this work. Although some related works published very promising results based on MRCGs, they showed mediocre performance in this work. If this was due to a erroneous implementation or a ill-suited combination of input features and classifier remains unknown.

The best results in this work were achieved by standard MFCCs. The number of MFCCs per frame was varied. The BLSTM-based system worked best with 40-dimensional MFCCs. The FSTDNN-based system on the other hand achieved the best result based on 64-dimensional MFCCs. Moreover, higher dimensional MFCCs were better suited for distinguishing speech from music, while lower dimensional MFCCs proved superior for distinguishing speech from music. This might be due to the trade-off between spectral and temporal attention when calculating different sizes of MFCCs from the same number of frequency bins. This could be further investigated by going one step further and varying the parameters of the DFT.

The two proposed systems achieve comparable performance. The margins in-between them considering several metrics are small. However, the threshold for BLSTM-based system calculated on the AVA validation data showed better generalization capability than it's counterpart for the FSTDNN-based system. The FSTDNN-based system on the other hand was not only slightly superior in terms of general performance. It also was less dependent on the type of background noise than the BLSTM-based system.

The relatively small advantage of the FSTDNN-based system over the BLSTM-based system seems counter-intuitive. On the other hand, the results of the BLSTM-based system were already good. Thus, further improvements might be difficult to achieve. However, several contemporary works report their best results based on complex combinations of FSTDNN and RNN layers. Possibly, a more complex or deeper FSTDNN-based architecture below the BLSTM might yield further improvements.

Although the proposed systems are ultimately aimed at supplementing an ASR system they were not tested as part of one. This would be another interesting expansion of this work. As stated before, different types of VAD systems are more or less tolerable for an ASR system. The very high recall at a reasonable precision achieved by the FSTDNN-based system with adjusted threshold might be highly beneficial.

Also, the systems proposed in this work applied no smoothing, as it yielded no benefit with respect to the examined metrics. An ASR system however might profit from the continuous or even the discrete smoothing. The systems themselves have been prepared for use in an offline ASR system. Thus, given an ASR system with a proper API and a suitable training corpus this should be testable with reasonable effort. Still, it exceeded the scope of this work and thus has not been done yet.

In conclusion, this work yields two things. First, a comprehensive overview of the necessary considerations when building a VAD system and their effects on the system behavior. And second, instructions on how to build such a system with reasonable performance. For each aspect of such a system, this chapter provided not only the findings of this work, but also open questions and expansion possibilities.

8 Conclusion

In this work, VAD on the audio tracks of movies was explored. The necessary preprocessing was described for both standard MFCCs and the recently introduced MRCGs. Both LSTMs and FSTDNNs were outlined. A comprehensive description of the training process for deep neural networks was provided. This included gradient descent, backpropagation, cross-entropy loss as objective function and the use of schedulers. Notably, SMORMS3, an alternative to the popular Adam scheduler was described. Different smoothing techniques were outlined as well. An overview of the metrics used in this work was provided.

VAD relies heavily on corpora of training audio and corresponding annotations. An overview over several corpora of interest was provided. The decision to use the AVA corpus for training and validation and the Hollywood corpus for evaluation was justified. These were described in detail.

A comprehensive overview of works on VAD was provided. Starting with the historical roots of both VAD and speech/music classification, the development leading to the modern setup of input features and a NN classifier was traced. Contemporary works of interest, assorted by used input features, were listed and their distinctive characteristics described.

The proposed systems were described in detail. They consist of a relatively standard preprocessing step, yielding MFCCs and a NN-based classifier. Several different features, as well as different dimensionalities of the MFCCs were tested. For the classification, several different architectures were analyzed. Two best performing models, one based solely on BLSTM layers, and one based on a mixture of BLSTM and FSTDNN layers, were determined. For these a thorough search of optimal combinations of input features and parameter combinations was conducted. Different smoothing techniques were also examined, and an overview of the used technology and software was provided.

The experiments leading to the proposed systems were described in detail. The performance of different combinations of input features and NN architectures is listed. For both proposed models the effect of varying the number of layers, hidden units and, in case of the FSTDNN, filters was explored and the results reported. The effects of different smoothing techniques are listed.

Two systems were chosen for further experiments. One consisted of four layers of BLSTMs with 1024 hidden units each. 40-dimensional MFCCs were used as input features. The two directions of the BLSTM were summed, and a feedforward network projected them onto four outputs. A softmax function was used to generate a probability distribution.

The second system consisted of four blocks of gated FSTDNNs with increasing filter size and dilation. Each FSTDNN featured 32 filters. The results of the four blocks were concatenated and fed to a BLSTM block similar to the first system. It featured 256 hidden units per layer, otherwise it was identical to the first network. This system used 64-dimensional MFCCs as input features. The second system performed slightly better.

Several smoothing techniques were tested on the outputs of both systems. None led to an increase in performance, thus the subsequent experiments were conducted without applying any smoothing. However, if one of the systems was used as pre-processor to an ASR system, some smoothing might improve overall performance.

The two proposed systems were then evaluated against other contemporary VAD systems. This was done based on a corpus of four Hollywood movies. A coarse analysis of the systems performances against the best systems in literature is provided. Two works published in-depth results on the four movies, using these a fine-grained analysis could be conducted as well. Furthermore a separated test set of the training data was used to examine, how the different combinations of systems and features handled background noise and music respectively.

It was shown that both systems performed on par with the best contemporary systems from literature. The in-depth analysis showed especially the recall of the BLSTM-based system to be very good, while preserving a competitive precision. The AUCs of both proposed systems outperformed all reported ones by a substantial margin.

Finally, the achieved results were discussed and set into perspective. The most important characteristics of the proposed systems, and their effects on the performance, were pointed out. The advantages and shortcomings compared to other contemporary systems were examined. Ideas on how to improve or use the proposed systems were contemplated.

VAD in the presence of music remains a difficult problem, but the tools to find a satisfactory solution for many applications are in place. Further research is confronted with the same decision as before, focus on the input features or the classifier. While not introducing new or substantially enhanced features, this work provided some insights on the interactions of input features and classifiers. Two NN models, capable of working as a robust classifier in a VAD system were presented and described in detail. Thus, a contribution to VAD research was made. By advancements in both areas, input features and classifiers, substantially better systems are to be expected in the future.

Acronyms

ASR automatic speech recognition. 3, 15, 16, 21–23, 26–28, 30, 33, 39, 46, 48, 49, 51, 52, *Glossary:* automatic speech recognition

AUC area under the curve. 19, 27, 31, 35–38, 41, 42, 44, 48, 49, *Glossary:* area under the curve

BLSTM bi-directional long short-term memory network. iv, vii, 4, 12, 26, 28, 31–33, 35–45, 47–49, *Glossary:* bi-directional long short-term memory network

BRNN bi-directional RNN. 12, 49, *Glossary:* bi-directional RNN

CEL cross-entropy loss. 16, 36, 49, *Glossary:* cross-entropy loss

CG-LSTM coordinated-gate LSTM. 27, 49, *Glossary:* coordinated-gate LSTM

DCT discrete cosine transform. 27, 34, 49, *Glossary:* discrete cosine transform

DFT discrete fourier transform. 7, 8, 45

DNN deep neural network. 24, 26–30, 40, 49, *Glossary:* deep neural network

EER equal error rate. 27, 33, 42, 49, *Glossary:* equal error rate

FFN feedforward network. 49, 52, *Glossary:* feedforward network

FNR false negative rate. 42, 49, 52, *Glossary:* false negative rate

FPR false positive rate. 42, 49, 52, *Glossary:* false positive rate

FSTDNN frequency shifting time delay neural network. iv, 11, 26–30, 32, 33, 36–49, *Glossary:* frequency shifting time delay neural network

HMM Hidden Markov Model. 26, 49, *Glossary:* Hidden Markov Model

LSTM long short-term memory network. 11, 12, 26–29, 47, 49, 51, *Glossary:* long short-term memory network

MFCCs Mel-frequency cepstral coefficients. iv, 4, 7, 8, 24–29, 31, 32, 34–37, 44, 45, 47–49, 51, *Glossary:* Mel-frequency cepstral coefficients

MLP multilayer perceptron. 10, 26, 29, 49, *Glossary:* multilayer perceptron

MRCGs multi-resolution cochleagrams. 4, 7, 8, 25, 26, 29, 30, 34–36, 45, 47, 50, *Glossary:* multi-resolution cochleagrams

NN neural network. i, 4, 9–13, 26–28, 31, 33, 45, 47, 48, 50–53, *Glossary:* neural network

RNN recurrent neural network. 11, 12, 14, 27, 45, 49–52, *Glossary:* recurrent neural network

ROC receiver operating characteristic. 18, 19, 39, 50, 51, *Glossary:* receiver operating characteristic

SAD speech activity detection. 3, 4, 22, 23, 27, 50, *Glossary:* speech activity detection

SGD stochastic gradient descent. 13, 14, 50, *Glossary:* stochastic gradient descent

SMC speech/music classification. 27, 47, 50, *Glossary:* speech/music classification

SMORMS3 squared mean over root mean squared cubed. 12, 14, 27, 47, 50, *Glossary:* squared mean over root mean squared cubed

SNR signal to noise ratio. 24, 50, *Glossary:* signal to noise ratio

TDNN time delay neural network. 11, 29, 38, 50, 52, *Glossary:* time delay neural network

VAD voice activity detection. iii, iv, 3–5, 7, 11, 16–18, 21–31, 40, 42, 46–48, 50–53, *Glossary:* voice activity detection

Glossary

Adam A popular optimization algorithm used during NN training. See section 2.3.4 for details. 27, 47

area under the curve The integral of the ROC curve. See section 2.5.6 for details. 19, 49

automatic speech recognition The transcription of an audio recording of speech into the corresponding written representation by a machine or software. 3, 15, 49

bi-directional long short-term memory network A special case of LSTM. Bidirectional in the sense of processing a sequence not only in the correct temporal order but also reversed. See section 2.2.4 for details. iv, vii, 12, 49

bi-directional RNN A pair of RNNs that processes a sequence in both regular and inverted order. 12, 49

clipping In the context of VAD, clipping is the discarding of frames containing speech because of a false negative VAD decision. Most applications of VAD are hurt severely by clipping, while some superfluous frames are usually acceptable. 16, 52

cochleagram A feature aimed at imitating human perception of sound. See section 2.1.2 for details. 8, 35, 36

coordinated-gate LSTM A specialized type of LSTM introduced by [GG18]. See section 4.2 for details. 27, 49

cross-entropy loss A loss function based on the dissimilarity between two probability distributions. See section 2.3.3 for details. 16, 49

deep neural network A type of NN. Originally any NN with at least one hidden layer. Recently mostly used when several hidden layers are present. 24, 49

discrete cosine transform A linear transform used widely for the computation of MFCCs. See section 2.1.1 for details. 27, 49

endpoint detection The detection of boundaries between speech and non-speech segments instead of either record-wise or frame-wise classification. Especially important for ASR and related tasks. 26

equal error rate The FPR and FNR if a threshold is fixed such that both are equal. See section 2.5.5 for details. 27, 49

false negative rate The rate at which frames predicted not to contain voice activity actually do so. See section 2.5.5 for details. 42, 49

false positive rate The rate at which frames predicted to contain voice activity actually do not. See section 2.5.5 for details. 42, 49

feedforward network A simple type of multilayer NN. See 2.2.1 for details. 49, 52

frame An extract from an audio stream. Usually short enough to consider speech signals stable within, about 10 ms is a common duration. 4

frequency shifting time delay neural network A variant of the TDNN architecture shift-invariant not only along the time but also the frequency axis. See section 2.2.3 for details. iv, 11, 49

hangover Frames added to a talkspurt deliberately even though not considered speech by the decision rule. Used to avoid clipping. 26

Hidden Markov Model A statistical Markov model wherein the states are unobservable so the state sequence needs to be derived from the output of these states. Historically very important to ASR and somewhat important to VAD. 26, 49

Kullback-Leibler divergence A measurement of of how different a probability distribution is from a reference probability distribution. Also called relative entropy. 26

long short-term memory network A type of RNN featuring four internal gates per cell for deciding what information to save, what to forget and what to put out. See section 2.2.4 for details. 11, 49

Mel-frequency cepstral coefficients Most commonly used features for neural network based VAD. See section 2.1.1 for details. iv, 4, 7, 49

minibatch A small set of training samples during stochastic gradient descent. See section 2.3.1 for details. 13, 28, 36, 53

multilayer perceptron A NN consisting of at least three layers, i.e. featuring at least one hidden layer. Mostly referring to feedforward networks (FFNs), but sometimes used for other types of NNs as well. 10, 49

multi-resolution cochleagrams A novel feature introduced by [CWW14]. See section 2.1.2 for details. 4, 7, 50

neural network A computer system consisting of a collection of nodes and their connections. The nodes may all behave the same or differently in groups. Their number is arbitrary, reaching into the billions. They might be arranged om one or multiple layers with arbitrary wiring complexity. i, 9, 50

perceptron First attempt at mimicking a neurons' behavior in computers. Important predecessor to modern NNs. 26

receiver operating characteristic Performance of a binary classifier depending on its classification threshold. See section 2.5.6 for details. 18, 50

recurrent neural network A type of NN where individual cells contain an internal state representing some combination of the previous states and inputs. Utilized mainly to process sequences of variable length. 11, 50

signal to noise ratio In speech processing the ratio between the signal power and the noise power. In VAD the ratio between the power of the speech signal and the power of the background noise. 24, 50

softmax Function used to project a vector of real values onto a probability vector (entries are between zero and one and add up to one). See section 2.2.2 for details. 31

soundtrack The entirety of music used in a movie. Consists of the score, specifically written for the movie and other, preexisting, compositions. 22

speech activity detection The detection of intelligible speech in an audio signal. 3, 22, 50

speech/music classification The process of deciding whether a given audio recording or excerpt is containing either speech or music. Depending on the subsequent task, mixed frames may be considered to belong to either of the classes or one or more specialized classes are possible. 27, 47, 50

speech/music discrimination The process of deciding whether a given recording or frame of an audio signal contains speech and/or music. 3, 5, 21

sprachraum Linguistic term for a geographical region wherein the same first language or group of languages is spoken. 3

squared mean over root mean squared cubed A training scheduler. See section 2.3.4 for details. 12, 50

stochastic gradient descent A variant of gradient descent, using minibatches. See section 2.3.1 for details. 13, 50

talkspurt Continuous segment of speech between two pauses. Term is mostly used when referring to digital telephony. 26, 52

time delay neural network A type of neural network used for audio tasks. Its most important characteristic is its invariance to shifts in the time domain. See section 2.2.3 for details. 11, 50

voice activity detection The detection of human voices in an audio signal. Includes unintelligible sounds like sighs. iii, iv, 3, 7, 50

zero crossing rate A measurement of how often a signal changes its sign. Mainly used for the detection of fricatives and percussive sounds in music detection. 25, 27

Bibliography

[19] (Jun. 10, 2019). "Implementing the SMORMS3 optimizer in PyTorch," Personalized TV on single-board computers, [Online]. Available: https://raberrytv.wordpress.com/2019/06/10/implementing-the-smorms3-optimizer-in-pytorch/.

[20] *Anaconda Software Distribution*. Anaconda Inc., 2020, Publication Title: Anaconda Documentation Version Number: Vers. 2-2.4.0.

[ALB93] P. Alexandre, P. Lockwood, and J. Boudy, "Evaluation of car noise reduction/compensation techniques for digit recognition in a speaker-independent context," presented at the 3rd European Conference on Speech Communication and Techology EUROSPEECH, Berlin, Germany: ISCA, Sep. 1993.

[AO77] T. Araseki and K. Ochiai, "Speech signal presence detector," U.S. Patent 4001505A, Library Catalog: Google Patents, Jan. 4, 1977.

[AR76] B. Atal and L. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 3, pp. 201–212, Jun. 1976, Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing.

[CBS+15] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 577–585.

[CG01] W. Chou and L. Gu, "Robust singing detection in speech/music discriminator design," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, ISSN: 1520-6149, vol. 2, May 2001, 865–868 vol.2.

[Cho18] A. Choné. (2018). "Computing MFCCs voice recognition features on ARM systems," Computing MFCCs voice recognition features on ARM systems, [Online]. Available: https://medium.com/linagoralabs/computing-mfccs-voice-recognition-features-on-arm-systems-dae45f016eb6.

[Cho20] K. Chowdhary, *Fundamentals of Artificial Intelligence*. New Delhi: Springer India, 2020.

[CLS+17] S.-Y. Chang, B. Li, T. N. Sainath, G. Simko, and C. Parada, "Endpoint detection using grid long short-term memory networks for streaming speech recognition," in *Interspeech 2017*, ISCA, Aug. 20, 2017, pp. 3812–3816.

[CLS+18] S.-Y. Chang, B. Li, G. Simko, T. N. Sainath, A. Tripathi, A. van den Oord, and O. Vinyals, "Temporal modeling using dilated convolution and gating for voice-activity-detection," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB: IEEE, Apr. 2018, pp. 5549–5553.

[CRE+18] S. Chaudhuri, J. Roth, D. P. W. Ellis, A. Gallagher, L. Kaver, R. Marvin, C. Pantofaru, N. Reale, L. G. Reid, K. Wilson, and Z. Xi, "AVA-speech: A densely labeled dataset of speech activity in movies," *arXiv:1808.00606 [cs, eess]*, Aug. 23, 2018. arXiv: 1808.00606.

[CWW14] J. Chen, Y. Wang, and D. Wang, "A feature study for classification-based speech separation at low signal-to-noise ratios," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1993–2002, Dec. 2014, Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[DC17] D. Doukhan and J. Carrive, "Investigating the use of semi-supervised convolutional neural network models for speech/music classification and segmentation," May 2017.

[DM80] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980, Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing.

[DSVM10] D. Dean, S. Sridharan, R. Vogt, and M. Mason, "The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms," *The QUT-NOISE-TIMIT Corpus for the Evaluation of Voice Activity Detection Algorithms*, pp. 3110–3113, Sep. 2010.

[Eur03] European Telecommunications Standards Institute, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front-end feature extraction algorithm; compression algorithms," Sep. 2003.

[EWSS13] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life voice activity detection with LSTM recurrent neural networks and an application to hollywood movies," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada: IEEE, May 2013, pp. 483–487.

[Faw06] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[FD67] R. F. J. Filipowsky and F. S. Davidson, "Voice-data multiplexing system for transmitting data during pauses in the voice signals," U.S. Patent 3311704A, Library Catalog: Google Patents, Mar. 28, 1967.

[Fit10] D. FitzGerald, "Harmonic/percussive separation using median filtering," p. 4, 2010.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[GG18] G. Gelly and J.-L. Gauvain, "Optimization of RNN-based speech activity detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 646–656, Mar. 2018.

[GLF+92] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Acoustic-phonetic continuous speech corpus CD-ROM," Nov. 1992.

[Goo20] Google Developers. (2020). "WebRTC," WebRTC, [Online]. Available: `https://webrtc.org/`.

[Gra12] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, ser. Studies in Computational Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 385.

[Gru82] J. Gruber, "A comparison of measured and calculated speech temporal parameters relevant to speech activity detection," *IEEE Transactions on Communications*, vol. 30, no. 4, pp. 728–738, Apr. 1982, Conference Name: IEEE Transactions on Communications.

[HFH+09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, Nov. 16, 2009.

[HM93] J. Haigh and J. Mason, "Robust voice activity detection using cepstral features," in *IEEE Region 10 International Conference on Computers, Communications and Automation Proceedings of TENCON '93*, vol. 3, Oct. 1993, 321–324 vol.3.

[Hoc98] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 2, pp. 107–116, Apr. 1998.

[HS97] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1, 1997.

[HSN19] R. Hebbar, K. Somandepalli, and S. Narayanan, "Robust speech activity detection in movie audio: Data resources and experimental evaluation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom: IEEE, May 2019, pp. 4105–4109.

[HZRS15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385 [cs]*, Dec. 10, 2015. arXiv: `1512.03385`.

[IW90] J. B. H. Ii and A. Waibel, "Connectionist architectures for multi-speaker phoneme recognition," 1990.

[Jan77] J. A. Jankowski, "Digital voice switch," U.S. Patent 4052568A, Library Catalog: Google Patents, Oct. 4, 1977.

[JASJ17] I. Jang, C. Ahn, J. Seo, and Y. Jang, "Enhanced feature extraction for speech detection in media audio," in *Interspeech 2017*, ISCA, Aug. 20, 2017, pp. 479–483.

[KB17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980 [cs]*, Jan. 29, 2017. arXiv: 1412.6980.

[KH18] J. Kim and M. Hahn, "Voice activity detection using an adaptive context attention model," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1181–1185, Aug. 2018.

[KLW19] U. Kamath, J. Liu, and J. Whitaker, *Deep Learning for NLP and Speech Recognition*. Cham: Springer International Publishing, 2019.

[KRH04] J. Ka˘cur, G. Rozinaj, and S. Herrera-Garcia, "SPEECH SIGNAL DETECTION IN a NOISY ENVIRONMENT USING NEURAL NETWORKS AND CEPSTRAL MATRICES," vol. 55, no. 5, p. 7, 2004.

[KSI18] T. Kosaka, I. Suga, and M. Inoue, "Improving voice activity detection for multimodal movie dialogue corpus," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, Nara: IEEE, Oct. 2018, pp. 481–484.

[LJZ77] R. E. LaMarche, C. J. M. Jr, and T. J. Zebo, "Digital speech detector," U.S. Patent 4028496A, Library Catalog: Google Patents, Jun. 7, 1977.

[LMHK20] Y. Lee, J. Min, D. K. Han, and H. Ko, "Spectro-temporal attention-based voice activity detection," *IEEE Signal Processing Letters*, vol. 27, pp. 131–135, 2020.

[LU86] H. Lee and C. Un, "A study of on-off characteristics of conversational speech," *IEEE Transactions on Communications*, vol. 34, no. 6, pp. 630–637, Jun. 1986, Conference Name: IEEE Transactions on Communications.

[LWH90] K. J. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol. 3, no. 1, pp. 23–43, Jan. 1990.

[LWS15] B. Lehner, G. Widmer, and R. Sonnleitner, "Improving voice activity detection in movies," Sep. 2015.

[mcd20] mcdermottLab, *mcdermottLab/pycochleagram*, original-date: 2018-05-01T22:11:44Z, Jun. 14, 2020.

[MMG19] B. Meléndez-Catalán, E. Molina, and E. Gómez, "Open broadcast media audio from TV: A dataset of TV broadcast audio with relative music loudness annotations," *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, pp. 43–51, Aug. 12, 2019.

[MP76] G. Monti and I. Poretti, "Method of and means for detecting voice frequencies in telephone system," U.S. Patent 3985956A, Library Catalog: Google Patents, Oct. 12, 1976.

[MSW18] M. Müller, S. Stüker, and A. Waibel, "Enhancing multilingual graphemic RNN based ASR systems using phone information," *29. Konferenz Elektronische Sprachsignalverarbeitung ESSV*, 2018.

[NVF20] NVIDIA, P. Vingelmann, and F. H. Fitzek, *CUDA, release: 10.2.89*. 2020.

[PGM+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035.

[PNHR88] R. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, "AN EFFICIENT AUDITORY FILTERBANK BASED ON THE GAMMATONE FUNCTION," p. 34, Jan. 1988.

[RHW86] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, "Learning representations by back-propagating errors," p. 4, 1986.

[RLY13] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on YouTube using deep neural networks," Jan. 2013.

[RMC07] J. Ramirez, J. M., and J. C., "Voice activity detection. fundamentals and speech recognition system robustness," in *Robust Speech Recognition and Understanding*, M. Grimm and K. Kroschel, Eds., I-Tech Education and Publishing, Jun. 1, 2007.

[RSB+04] J. Ramirez, J. Segura, C. Benitez, A. de la Torre, and A. Rubio, "A new kullback-leibler VAD for speech recognition in noise," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 266–269, Feb. 2004, Conference Name: IEEE Signal Processing Letters.

[RYGS06] J. Ramirez, P. Yelamos, J. Gorriz, and J. Segura, "SVM-based speech endpoint detection using contextual speech features," *Electronics Letters*, vol. 42, no. 7, pp. 426–428, Mar. 2006, Conference Name: Electronics Letters.

[Sas07] Y. Sasaki, "The truth of the f-measure," Oct. 26, 2007.

[Sav89] M. Savoji, "A robust algorithm for accurate endpointing of speech signals," *Speech Communication*, vol. 8, no. 1, pp. 45–60, Jun. 1989.

[SCP15] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv:1510.08484 [cs]*, Oct. 28, 2015. arXiv: `1510.08484`.

[SDD19] S. Sen, A. Dutta, and N. Dey, *Audio Processing and Speech Recognition: Concepts, Techniques and Research Overviews*, ser. SpringerBriefs in Applied Sciences and Technology. Singapore: Springer Singapore, 2019.

[Sim15] Simon Funk. (Apr. 20, 2015). "RMSprop loses to SMORMS3 - beware the epsilon!" [Online]. Available: `https://sifter.org/~simon/journal/20150420.html`.

[SPS07] K. Seyerlehner, T. Pohle, and M. Schedl, "Automatic music detection in television productions," *Automatic Music Detection in Television Productions.*, 2007.

[SS12] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," *Speech Communication*, vol. 54, no. 4, pp. 543–565, May 2012.

[SS97] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, ISSN: 1520-6149, vol. 2, Apr. 1997, 1331–1334 vol.2.

[SS98] J. Sohn and W. Sung, "A voice activity detector employing soft decision based noise spectrum adaptation," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, ISSN: 1520-6149, vol. 1, May 1998, 365–368 vol.1.

[SSR01] J. Stadermann, V. Stahl, and G. Rose, "Voice activity detection in noisy environments," p. 4, 2001.

[SSVC17] D. A. Silva, J. A. Stuchi, R. P. V. Violato, and L. G. D. Cuozzo, "Exploring convolutional neural networks for voice activity detection," in *Cognitive Technologies*, A. Paradisi, A. Godoy Souza Mello, F. Lira Figueiredo, and R. Carvalho Figueiredo, Eds., Series Title: Telecommunications and Information Technology, Cham: Springer International Publishing, 2017, pp. 37–47.

[SZ15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556 [cs]*, Apr. 10, 2015. arXiv: 1409.1556.

[TGY16] S. Tong, H. Gu, and K. Yu, "A comparative study of robustness of deep learning approaches for VAD," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, ISSN: 2379-190X, Mar. 2016, pp. 5695–5699.

[TS18] J. Tracey and S. Strassel, "VAST: A corpus of video annotation for speech technologies," May 2018.

[VSP+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv:1706.03762 [cs]*, Dec. 5, 2017. arXiv: 1706.03762.

[War17] M. Warsewa, "Monaural Speech Separation with Deep Neural Networks," Sep. 2017.

[WHH+89] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar. 1989, Conference Name: IEEE Transactions on Acoustics, Speech, and Signal Processing.

[YISK16] R. Yasuhara, M. Inoue, I. Suga, and T. Kosaka, "Large-scale multimodal movie dialogue corpus," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction - ICMI 2016*, Tokyo, Japan: ACM Press, 2016, pp. 414–415.

[ZSSP16] R. Zazo, T. N. Sainath, G. Simko, and C. Parada, "Feature learning with raw-waveform CLDNNs for voice activity detection," presented at the Interspeech 2016, Sep. 8, 2016, pp. 3668–3672.

[ZW16] X.-L. Zhang and D. Wang, "Boosting contextual information for deep neural network based voice activity detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 252–264, Feb. 2016, Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.