An End-to-End Multimodal Voice Activity Detection Using WaveNet Encoder and Residual Networks

Ido Ariav and Israel Cohen

***Abstract*—Recently, there has been growing use of deep neural networks in many modern speech-based systems such as speaker recognition, speech enhancement, and emotion recognition. Inspired by this success, we propose to address the task of voice activity detection by incorporating auditory and visual modalities into an end-to-end deep neural network. We evaluate our proposed system in challenging acoustic environments including high levels of noise and transients, which are common in real life scenarios. Our multimodal setting includes a speech signal captured by a microphone and a corresponding video signal capturing the speaker's mouth region. Under such difficult conditions, robust features need to be extracted from both modalities in order for the system to accurately distinguish between speech and noise. For this purpose, we utilize a deep residual network (ResNet), to extract features from the video signal, while for the audio modality we employ a variant of WaveNet encoder for feature extraction. The features from both modalities are fused using multimodal compact bilinear pooling (MCB) to form a joint representation of the speech signal. To further encode the temporal information we feed the fused signal to a Long Short-Term Memory (LSTM) network and the system is then trained in an end-to-end supervised fashion. Experimental results demonstrate the improved performance of the proposed end-to-end multimodal architecture compared to unimodal variants for voice activity detection (VAD). Upon the publication of this paper, we will make the implementation of our proposed models publicly available at https://github.com/iariav/End-to-End-VAD and https://israelcohen.com.**

***Index Terms*—Audio-visual speech processing, voice activity detection, deep neural networks, wavenet**

I. INTRODUCTION

Voice activity detection constitutes an essential part of many modern speech-based systems, and its applications can be found in various domains. A partial list of such domains includes speech and speaker recognition, speech enhancement, dominant speaker identification, and hearing-improvement devices. In many cases, voice activity detection is used as a preliminary block to separate the segments of the signal that contain speech from those that contain only noise and interferences, thus enabling the overall system to, e.g., perform speech recognition only on speech segments, or change the noise reduction method between speech/noise segments.

Traditional methods of voice activity detection mostly rely on the assumption of quasi-stationary noise, i.e., the noise spectrum changes at a much lower rate than the speech signal. One group of such methods are those based on simple acoustic features such as zero-crossing rate and energy values in short time intervals [1], [2], [3]. More advanced methods are model-based methods that focus on estimating a statistical model for the noisy signal [4], [5], [6], [7]. The performance of such methods usually significantly deteriorates in the presence of even moderate levels of noise. Moreover, they cannot correctly model highly non-stationary noise and transient interferences, which are common in real life scenarios and are within the main scope of this study, since the spectrum of transients, similarly to the spectrum of speech, often rapidly varies over time [8].

Apart from the statistical approaches noted above, more recent methods have been developed using machine learning techniques [9], [10], and more specifically, using deep neural networks. In recent years, deep neural networks achieved groundbreaking improvements on several pattern recognition benchmarks in areas such as image classification [11], speech and speaker recognition [12] and even multimodal tasks such as visual question answering [13]. Deep networks were successfully used to extract useful signal representations from raw data, and more specifically, several studies have also shown the favorable property of deep networks to model the inherent structure contained in the speech signal [14]. Deep neural networks were recently utilized in several modern voice activity detectors; Zhang and Wu [15] proposed to extract a set of predefined acoustic features, e.g., Mel Frequency Cepstral Coefficients (MFCC), from a speech signal and then feed these features to a deep-belief network (DBN) in order to obtain a more meaningful representation of the signal. They then used a linear classifier to perform speech detection. Thomas et al. [16] fed a convolutional neural network (CNN) with the log-Mel spectrogram together with its delta and delta-delta coefficients and trained the CNN to perform voice activity detection.

Despite showing improved performance compared to traditional methods, these networks classify each time frame independently, thus ignoring existing temporal relations between consecutive time frames. To alleviate this issue, several studies have suggested methods for modeling temporal relations between consecutive time frames [17]. More modern methods rely on recurrent neural networks (RNN) to incorporate previous inputs into the classification process, thus utilizing the signal's temporal information [18], [19], [20]. Hughes and Mierle [21] extracted Perceptual Linear Prediction (PLP) features from a speech signal and fed them to a multi-

The authors are with the Andrew and Erna Viterbi Faculty of Electrical Engineering, The Technion-Israel Institute of Technology, Haifa 3200003, Israel
(e-mail: idoariav@tx.technion.ac.il;icohen@ee.technion.ac.il)

This research was supported by the Israel Science Foundation (grant No. 576/16) and the ISF-NSFC joint research program (grant No. 2514/17).

layered RNN with quadratic polynomial nodes to perform speech detection. Lim et al. [22] proposed to transform the speech signal using a short-time Fourier transform and use a CNN to extract high-level representation for the signal. This new representation was then fed to an LSTM to exploit the temporal structure of the data. These methods however still mostly rely on hand-crafted audio features, and often misclassify frames that contain both speech and transients as non-speech frames, since transients often appear more dominant than speech.

Although most of the current work on voice activity detection concentrates around the subject's audio signal, recent methods proposed to make use of other modalities, such as visual information, to improve the voice detection [23], [24], [25], [26], [27], [28], [29], [30], [31]. The advantages of using a multimodal setting are most prominent when dealing with demanding acoustic environments, where high levels of acoustic noise and transient interferences are present since the video signal is entirely invariant to the acoustic environment. Therefore, proper incorporation of the video signal can significantly improve voice detection, as we show in this paper. Ngiam et al. [32] proposed a Multimodal Deep Autoencoder for feature extraction from audio and video modalities. A bimodal DBN was used for the initialization of the deep autoencoder, and then the autoencoder was fine-tuned to minimize the reconstruction error of the two modalities. Zhang et al. [33] used a CNN for classifying emotions in a multimodal setting. They applied two separate CNNs, the first operating on the mel-spectrogram of an audio signal and the second on a video recording of a subject's face. The features extracted using the two CNNs were concatenated and fed to a deep fully connected neural network to perform the classification. Dov et al. [34] proposed to obtain separate low dimensional representations of the audio and video signals using diffusion maps [35]. The two modalities were then fused by a combination of speech presence measures, which are based on spatial and temporal relations between samples of the signal in the low dimensional domain. In our previous work [36], we proposed a deep architecture comprised of a transient reducing autoencoder and an RNN for voice activity detection. Features were extracted from both modalities and fed to the transient reducing autoencoder which was trained to both reduce the effect of transients and merge the modalities. The output of the autoencoder was fed to an RNN that incorporated temporal data to the speech presence/absence classification.

The great majority of works described above still make use of commonly hand-crafted features in audio or visual modality. To alleviate the need for hand-crafted features, a few studies proposed to adopt an end-to-end approach and use the raw, unprocessed data as input while utilizing as little human apriori knowledge as possible [37]. The motivation behind this being that a deep network can ultimately automatically learn an intermediate representation of the raw input signal that better suits the task at hand which in turn leads to improved overall performance. Trigeorgis et al. [38] proposed an end-to-end model for emotion recognition from raw audio signal. A CNN was used to extract features from the raw signal which were then fed to an LSTM network in order to capture the temporal information in the data. Tzirakis et al. [39] recently proposed another multimodal end-to-end method for emotion recognition. Features were extracted separately from audio and video signals using two CNNs and then concatenated to form a joint representation, which in turn was fed to a multi-layered LSTM for classification. Hou et al. [40] proposed a multimodal end-to-end setting for speech enhancement. They used two CNNs to extract features from audio spectrograms and raw video, and these features were then concatenated and fed into several fully connected layers to produce an enhanced speech signal. This work, however, uses only simple concatenation to fuse the two modalities and does not utilize temporal information which we solve by incorporating LSTM. Petridis et al. [41] proposed an end-to-end approach for audio-visual speech recognition. They extracted features from each modality using a CNN and then fed these features to modality-specific RNN layers. The modalities were then fused by feeding the outputs of those RNNs to another RNN layer.

In this paper, we propose a deep end-to-end neural network architecture for audio-visual voice activity detection. First, we extract meaningful features from the raw audio and video signals; for the video signal we employ a ResNet-18 network [42] as a feature extractor, and for the audio signal we employ a variant of a WaveNet [43] encoder. Then, instead of merely concatenating the feature vectors extracted from the two modalities, as is common in most multimodal networks, we propose to fuse the two vectors into a new joint representation via a Multimodal Compact Bilinear (MCB) pooling [44] module, which was shown to efficiently and expressively combine multimodal features. The output of the MCB module is fed to several stacked LSTM layers in order to explore even further the temporal relations between samples of the speech signal in the new representation. Finally, a fully connected layer is used to perform the classification of each time-frame to speech/non-speech, and the entire network is trained in a supervised end-to-end manner. To the best of our knowledge, this is the first time that such an end-to-end approach is applied for voice activity detection.

The proposed deep end-to-end architecture is evaluated in the presence of highly non-stationary noises and transient interferences. Experimental results show the benefits of our multimodal end-to-end architecture compared to unimodal approaches, and the advantage of audio-video data fusion is thus demonstrated. Also, we demonstrate the effectiveness of the MCB module for modality fusion compared to a simple concatenation/multiplication of feature vectors.

The remainder of the paper is organized as follows. In Section II, we formulate the problem of voice activity detection and present our dataset. In Section III, we introduce the proposed multimodal end-to-end architecture. In Section IV, we demonstrate the performance of the proposed deep end-to-end architecture for voice activity detection. Finally, in Section V, we conclude and offer some directions for future research.

II. DATASET AND PROBLEM FORMULATION

A. Problem Formulation

Voice activity detection is a segmentation problem in which segments of a given speech signal are classified as sections that contain speech and sections that contain only noise and interferences. We consider a speech signal recorded via a single microphone and a video camera simultaneously, pointed at a front-facing speaker reading an article aloud.

Let $\mathbf{a} \in \mathbb{R}$ and $\mathbf{v} \in \mathbb{R}^{H \times W \times 3}$ be the audio and video signals, respectively, where H and W are the height and width of each video frame in pixels, respectively. For alignment of the two signals, we artificially divide $\mathbf{a}$ into frames of length M and denote each video/audio frame with subscript n.

We use the clean audio signal $\mathbf{a}$ to label each time frame n according to the presence or absence of speech, and then assign each frame in $\mathbf{a}$ and $\mathbf{v}$, with the appropriate label. Our proposed architecture performs a frame-by-frame classification for voice activity detection, however, as part of the classification process of each frame, we also take into account T past frames. For this reason, we construct from $\mathbf{a}$ and $\mathbf{v}$ a dataset of N overlapping sequences of audio and video by concatenating consecutive frames into sequences of length T. We denote by $\mathbf{S}_a^i$ and $\mathbf{S}_v^i$, respectively, the i^{th} audio and video sequences. Since we only use past frames in the classification process, each sequence is labeled according to the label of the last frame in the sequence. That is, the sequence $\mathbf{S}_v^i$ containing the video frames $\{\mathbf{v}_{i-14}, \mathbf{v}_{i-13}, \mathbf{v}_{i-12}, ..., \mathbf{v}_i\}$ is assigned the label given to $\mathbf{a}_i$, the i^{th} frame of the audio signal $\mathbf{a}$. Each $\mathbf{S}_a^i$ is then contaminated with background noises and transient interferences, as described in Section II-B.

The goal in this study is to classify each frame n as a speech or non-speech frame.

B. Dataset

We evaluate the proposed deep end-to-end architecture for voice activity detection using the dataset presented in [34], [36]. The dataset includes audio and video recordings of 11 speakers reading aloud an article. The speakers are instructed to make natural pauses every few sentences so that the intervals of speech and non-speech range from several hundred ms to several seconds in length. The video signal $\mathbf{v}$ comprises the mouth region of the speaker, cropped from the original recording. It is processed at 25 frames/s and each frame is 90×110 pixels in size. The audio signal is recorded at 8 kHz with an estimated SNR of ~ 25 dB, and we artificially divide the signal to frames, where each frame contains $M = 320$ audio samples without overlap. Each of the 11 recordings is 120 seconds long. Each clean audio recording $\mathbf{a}$ is normalized to the range $[-1, 1]$ and each video recording $\mathbf{v}$ is normalized to have zero mean and standard deviation 1 in each of the three R, G, and B channels. We refer the reader to [34] for more details on the creation of the dataset.

We divide the audio and video recordings, $\mathbf{a}$ and $\mathbf{v}$, to overlapping sequences, $\mathbf{S}_a^i$, $\mathbf{S}_v^i$, of length T frames. This procedure produces an overall of $\sim 33,000$ such sequences. Out of the 11 speakers, we randomly select 8 speakers for the training set, and the other 3 speakers were used as an evaluation set. Finally, our training set contains $\sim 24,000$ audio/video sequences and the evaluation set contains $\sim 9,000$ sequences.

During the construction of the dataset, we randomly add background noise and transient interferences to each clean audio sequence $\mathbf{S}_a^i$ in the evaluation set according to the following procedure outlined in Alg. 1. First, background noise is randomly selected from one of {white Gaussian noise, musical instruments noise, babble noise, none} and a transient interference is randomly selected from one of {door-knocks, hammering, keyboard typing, metronome, scissors, none}, all taken from [45]. Once a background noise and transient were selected randomly, we randomly choose an SNR in the range $[0, 20]$ and then $\mathbf{S}_a^i$ is contaminated with the selected noise and transient at the selected SNR. We denote the contaminated sequence as $\tilde{\mathbf{S}}_a^i$. This way, our evaluation set contains all possible combinations of background noises and transients at different SNR levels. For the training set we use a similar procedure for injecting noise and transients. However, instead of adding the noise only once during the construction of the dataset, we inject the noise in each iteration of the training process. This way, during the entire training process, a specific sequence $\mathbf{S}_a^i$ can be injected with different combinations of background noise, transient, and SNR level. Note that in addition to noise injection, augmenting the video signal or altering the speakers' voice signals according to the injected noise levels (Lombard effect), can be applied to the same dataset, but this will be explored in future work.

Algorithm 1 Inject Random Background Noise and Transient Interference

▷ *Init noises and transients lists:*
Noises : {white Gaussian noise, musical instruments noise, babble noise, none}
Trans : {door-knocks, hammering, keyboard typing, metronome, scissors, none}

▷ *For each sequence $\mathbf{S}_a^i$ in the dataset:*
for $i = 1 \rightarrow \#sequences\ in\ dataset$ **do**
  $L \leftarrow$ Length of sequence $\mathbf{S}_a^i$ in frames
  ▷ *randomly choose noises to inject:*
  $noise \leftarrow$ uniformly random noise from "Noises"
  $trans \leftarrow$ uniformly random transient from "Trans"
  $SNR \leftarrow$ uniformly random value from [0,20]
  if $noise$ is not "none" **then**
    $R_{noise} \leftarrow$ random sequence of length L from $noise$
    $R_{noise} \leftarrow R_{noise}/std(R_{noise})$ ▷ update R_{noise}'s SNR
    $R_{noise} \leftarrow R_{noise} \times (std(\mathbf{S}_a^i)/(10^{(SNR/20)})$
    $\mathbf{S}_a^i \leftarrow \mathbf{S}_a^i + R_{noise}$
  if $trans$ is not "none" **then**
    $R_{trans} \leftarrow$ random sequence of length L from $trans$
    $\mathbf{S}_a^i \leftarrow \mathbf{S}_a^i + 2 \times R_{trans}$
return $\mathbf{S}_a^i$ as $\tilde{\mathbf{S}}_a^i$

It is worth noting that even though we use the same speech recordings as in [36], [34], the dataset we construct from them is somewhat different and significantly more challenging. In our experiments, each sample of the evaluation set contains a different mixture of background noise, transient, and SNR. In contrast, [36], [34] contaminated their evaluation set with only one background noise and one transient at a predefined

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTSP.2019.2901195, IEEE Journal of Selected Topics in Signal Processing

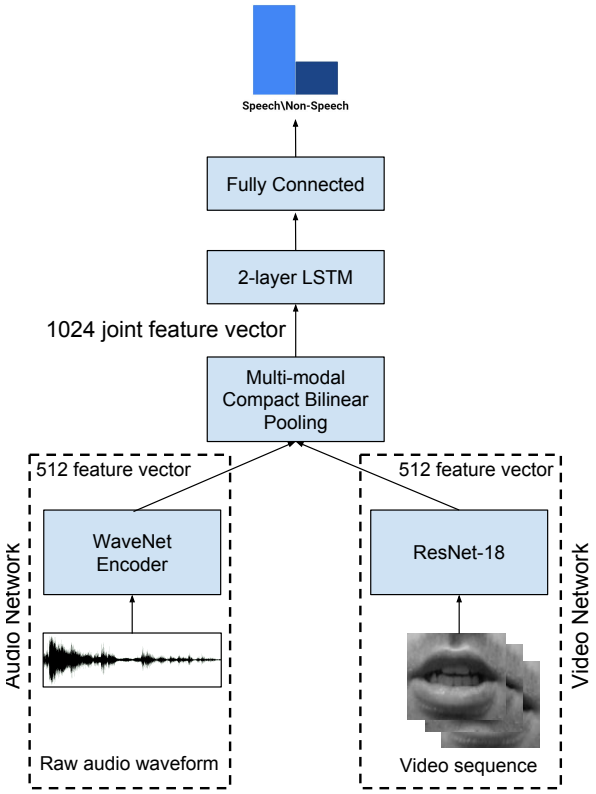

Fig. 1. Our proposed deep multimodal end-to-end architecture for voice activity detection.

SNR in each experiment.

III. DEEP MULTIMODAL END-TO-END ARCHITECTURE FOR VOICE ACTIVITY DETECTION

Voice activity detection becomes even more challenging in the presence of transients, which are typically more dominant than speech due to their short duration, high amplitudes and fast variations of the spectrum [8]. Specifically, audio features extracted from frames that contain both speech and transients are often similar to features extracted from frames that contain only transients, so that they are often wrongly classified as non-speech frames. To address this challenge, we introduce a deep end-to-end neural network architecture, in which meaningful features are learned from raw audio and video data. The overall system is trained in an end-to-end manner to predict speech presence/absence, and hence the learned features are those that maximize the classification accuracy. We propose to extract such features from both video and audio using two variants of known neural networks, ResNet and WaveNet, respectively, which we will now review. An overview of our deep multimodal end-to-end architecture for voice activity detection can be seen in Fig. 1.

A. Visual Network

In order to extract features from the raw video signal we use a deep residual network of 18 layers [42] denoted ResNet-18. Deep ResNets are formed by stacking several residual blocks of the form:

$$\mathbf{y}_k = F(\mathbf{x}_k, \mathbf{W}_k) + h(\mathbf{x}_k), \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output of residual block k, F is the residual block's function to be learned, and $h(\mathbf{x}_k)$ is either an identity mapping or a linear projection so that the dimensions of function F and the input $\mathbf{x}$ will match. The first layer of a ResNet-18 model is a 7×7 convolutional layer with 64 feature maps, and it is followed by a 3×3 max pooling layer. These two layers are followed by four residual blocks, where after each residual block a shortcut connection is added. Each residual block contains two convolutional layers of sizes 3×3, and the outputs of these residual blocks contain 64, 128, 256 and 512 feature maps respectively. After the last residual block, an average pooling layer is inserted followed by a fully connected layer performing the classification.

In order to use the ResNet-18 model to generate an embedding for the video sequence $\mathbf{S}_v^i$, we drop the last fully connected layer of the model and use the output of the average pooling layer as the video embedding which we denote $\mathbf{Z}_v$. The average pooling layer has 512 neurons, and thus the size of the produced embedding $\mathbf{Z}_v$ in feature space is 512 in length, for each frame of $\mathbf{S}_v^i$. $\mathbf{Z}_v$ has a temporal size of T, to match the temporal length of $\mathbf{S}_v^i$ so that the overall size of $\mathbf{Z}_v$ is $T\times512$. In order to avoid feeding each image in the sequence $\mathbf{S}_v^i$ to the network separately in a serial fashion, we concatenate all images into one batch, with T images, and feed this batch of images to the video network. The network then operates on all images in a parallel fashion, which provides substantial reduction of computation time.

It is worth noting that we also experimented with deeper residual models such as ResNet-50 and ResNet-101. However, they showed no improvement regarding the task of voice activity detection and some even experienced degradation in performance. This degradation can perhaps be explained by over-fitting of the model since these very deep models are usually used for tasks with hundreds or even thousands of classes whereas voice activity detection is a binary classification problem. These networks also produce longer embeddings, i.e., 2048 in size. We thus opted to use a shallower model with 18 layers, which also significantly reduces the memory consumption and computational load of the overall network.

B. Audio Network

In contrast to most previous machine learning works in audio processing, in which the first step is to extract hand-crafted features from the data, we propose to learn the feature extraction and classification steps in one jointly trained model for voice activity detection. The input to our audio network is the sequence of noisy raw audio signal frames $\tilde{\mathbf{S}}_a^i$. The feature extraction from the raw audio signal is performed by a WaveNet encoder, comprised of stacked residual blocks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTSP.2019.2901195, IEEE Journal of Selected Topics in Signal Processing

of dilated convolutions, which exhibit very large receptive fields, and which we will now review. The WaveNet encoder is designed in such a manner that enables it to better deal with long-range temporal dependencies that exist in the audio signal, than ordinary CNN or feed-forward networks.

1) Dilated Convolution: Convolutions are one of the main building blocks of modern neural networks. A convolution layer's parameters consist of a set of learnable filters $\{\mathbf{K}_q\}$, that have the same number of dimensions as the input they are convolved with. During the forward pass, these filters are convolved with the input, computing the dot product between the entries of the filter and the input to produce a new feature map. In most modern networks, relatively small filters are often used, thus each entry of the feature map has only a small receptive field of the input. In order to increase this receptive field, larger filters can be used, or many layers of small filters can be stacked, both at the expense of more burdensome computations. Here, we opted to use dilated convolutions to increase the receptive field of each feature map entry, without substantially increasing the computational cost.

In a dilated convolution, the filter $\mathbf{K}_q$ is applied over an area that is larger than its size by skipping input values with a predetermined dilation factor. E.g, in the 2-D case, a convolution operation between a 3×3 filter $\mathbf{K}_q$ with dilation factor of 2 and a 2-D input volume $\mathbf{I}$ at location (p, o) is given by:

$$(\mathbf{I} * \mathbf{K}_q)(p, o) = \sum_{l=-1}^{1} \sum_{m=-1}^{1} \mathbf{I}(p-2\,l, o-2\,m)\, \mathbf{K}_q(l+1, m+1) \tag{2}$$

where $*$ denotes the convolution operator. By skipping input values, a dilated convolution effectively operates on a larger receptive field than a standard convolution.

In order to increase the receptive field of all feature maps' entries even further, without increasing the computational load, several dilated convolutions can be stacked [46]. Notably, a block constructed of 10 dilated convolutions with exponentially increasing dilation factors of 1, 2, 4, . . . , 512, has a receptive field of size 1024 and can be considered a more efficient and discriminative non-linear counterpart of a 1×1024 regular convolution layer. We utilize this property in our implementation as described in Section III-B2.

2) WaveNet Encoder: WaveNet [43] is a powerful generative approach to probabilistic modeling of raw audio. Recalling the original WaveNet architecture described in [43], a WaveNet network is a fully convolutional neural network constructed by stacked blocks of dilated convolutions, where the convolutional layers in each block have exponentially growing dilation factors that allow the receptive field to also grow exponentially with depth and cover thousands of time-steps. A WaveNet model makes use of both residual [42] and parameterized skip connections throughout the network, which facilitates faster convergence of the model and enables the training of much deeper models. For more details on the original WaveNet architecture, we refer the readers to [43].

A WaveNet network is trained to predict the next sample of audio from a fixed-size input of prior sample values. In this paper, we use similar building blocks as in WaveNet to construct a WaveNet encoder that operates on a raw audio signal and, instead of predicting the next sample, it produces an embedding $\mathbf{Z}_a$ for each $\tilde{\mathbf{S}}_a^i$. This embedding $\mathbf{Z}_a$ is then used as the feature vector for the given audio sequence.

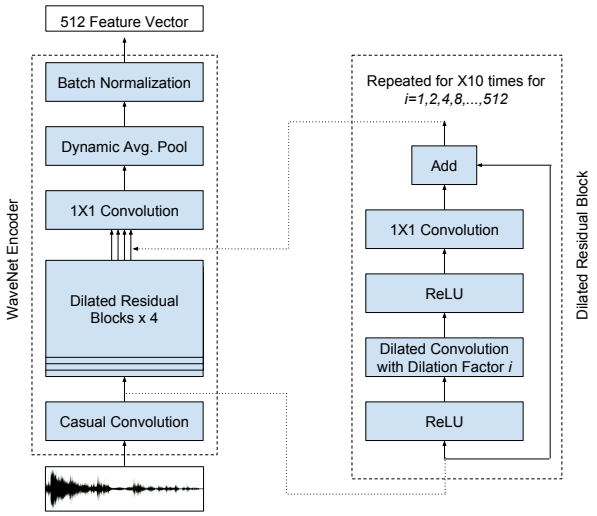

Fig. 2. Left - our WaveNet encoder architecture. Right - the structure of a dilated residual block.

Our implementation of a WaveNet encoder consists of a causal convolution layer followed by four stacked residual blocks, where each block is constructed by stacking 10 layers of dilated convolutions with exponentially growing dilation factors, ranging from 1 to 512 in each block. We found in our experiments that setting a fixed size of 32 channels for all convolution layers allows the model to be expressive enough while not making the network unnecessarily large. The output of all residual blocks is then aggregated and fed into a 1-D regular convolution with a kernel size of 1 and 512 channels so that the final dimension of $\mathbf{Z}_a$ in feature space is 512. We then finally apply an adaptive 1-D average pooling which operates on the temporal dimension, in order to further aggregate the activations of all residual blocks, and so that the temporal length of the embedding $\mathbf{Z}_a$ will be T, to match that of the video signal. Thus, the final size of $\mathbf{Z}_a$ is $T \times 512$. Throughout the network, we use 1-D filters of length 2 for all regular and dilated convolutions, and each convolution layer precedes a ReLU nonlinearity [47]. In Fig. 2 we show our WaveNet encoder overall architecture and the dilated residual block, which is stacked several times in the network.

Note that in contrary to the original WaveNet, we do not use μ-law compounding transformation to quantize the input, and instead, we operate on the raw 1-D signal. Moreover, early experiments did not show any noticeable advantage to the non-linearity used in [43], comprised of the element-wise multiplication of a tanh and sigmoid functions, over ReLU, so we opted to use the latter.

C. Multimodal Compact Bilinear Pooling

Once the embeddings for the audio and video signals, $\mathbf{Z}_a$ and $\mathbf{Z}_v$, are obtained, via the audio and video networks respectively, both embeddings are fused to form a joint representation that is then fed to the classification layers. Here, we propose to rely on multimodal compact bilinear pooling (MCB) to obtain this joint representation, rather than on simple concatenation of the embeddings.

Bilinear pooling is a fusion method for two vectors, $\mathbf{f} \in \mathbb{R}^{B_1}$ and $\mathbf{g} \in \mathbb{R}^{B_2}$, in which the joint representation is simply the outer product between the two vectors. This allows, in contrast to an element-wise product or simple concatenation, a multiplicative interaction between all elements of both vectors. Bilinear pooling models [48] have recently been used for fine-grained classification tasks [49]. However, their main drawback is their high dimensionality of $B_1 \times B_2$, which can be very large in most real-life cases and leads to an infeasible number of learnable parameters. For example, in our paper B_1 and B_2 are the lengths of the embeddings $\mathbf{Z}_a$ and $\mathbf{Z}_v$ in feature space, namely $B_1 = B_2 = 512$, which results in a joint representation of length 512^2. This inevitably leads to very high memory consumption and high computation times and can also result in over-fitting.

Here, we adopt ideas from [50] to the multimodal case for audio and visual modalities, and use MCB to fuse the two embeddings $\mathbf{Z}_a$ and $\mathbf{Z}_v$. As discussed in detail in [50], the multimodal compact bilinear pooling is approximated by projecting the joint outer product to a lower dimensional space, while also avoiding computing the outer product directly. This is accomplished via the count sketch projection function suggested in [51], denoted as Ψ, which is a method of projecting a vector $\mathbf{c} \in \mathbb{R}^m$ to a lower dimensional representation $\hat{\mathbf{c}} \in \mathbb{R}^d$ for $d < m$. Instead of applying Ψ directly on the outer product of the two embeddings, we follow [52] which showed that explicitly computing the outer product of the two vectors can be avoided since the count sketch of the outer product can be expressed as a convolution of both count sketches. Additionally, the convolution theorem states that a circular convolution of two discrete signals can be performed with lower asymptotic complexity by performing multiplication in the frequency domain (note that linear convolution in the time-domain may be replaced with a circular convolution by applying a proper zero-padding to the time-domain signals). Therefore, we project both embeddings $\mathbf{Z}_a$ and $\mathbf{Z}_v$ separately to a lower dimensional representation using Ψ and then calculate element-wise products of fast Fourier transforms (FFT) of the lower dimensional representations.

We apply MCB to $\mathbf{Z}_a$ and $\mathbf{Z}_v$ to form a joint representation of the two modalities, denoted $\mathbf{Z}_{av}$. We choose the MCB output size to be 1024 in feature space, and its temporal size is T so that the final size of $\mathbf{Z}_{av}$ is $T \times 1024$. We then apply batch normalization before feeding $\mathbf{Z}_{av}$ as input to the classification layers detailed in Section III-D. We note that MCB can easily be extended and remains effective for more than two modalities as the fusion of the modalities is achieved by element-wise product of FFTs. Another advantage to using MCB for vector fusion is being able to choose the desired size for the joint vector. In contrast, when feature vectors are fused by simple concatenation, element-wise multiplication or dot product, the joint representation is given by the sizes of the feature vectors. In MCB, the size of the joint representation can be selected according to performance or computational requirements.

In Section IV we demonstrate the effectiveness of fusing the two modalities with MCB compared to a simple concatenation of the embeddings.

D. Classification Layers

The joint embedding $\mathbf{Z}_{av}$ produced from the MCB module is fed to an LSTM block with two layers, each with 1024 cells, to explore even more temporal information embedded in the speech signal. We follow a "many-to-one" approach and feed only the last temporal response of the last LSTM layer, denoted $\mathbf{Y}_{av}$, to a fully connected layer with 1024 neurons, to match the size of the last LSTM layer. This fully conected layer is followed by another fully connected layer with just one neuron representing the output of the whole network. We apply a sigmoid activation function, so that the output of that final layer, denoted $\mathbf{Out}_{av}$, is constrained to the range of $0 - 1$ and can be considered as the probability for speech absence/presence.

For regularization purposes, and to avoide over-fitting, due to the large number of parameters in the network compared to the number of training examples, we use dropout [53] at several points in our network. We use a dropout with probability $p = 0.5$ at the last fully connected layer and dropout with probability $p = 0.2$ before and after the MCB module. We also use batch normalization on the outputs of the audio and video networks and the MCB module's output.

We summarize all the tensors in the final implementation of our multimodal end-to-end network and their sizes in Table I.

TABLE I
TENSOR NOTATIONS AND SIZES OF OUR FINAL MULTIMODAL NETWORK. THE FIRST DIMENSION IS ALWAYS THE TEMPORAL DIMENSION, AND THE REST ARE DIMENSIONS IN DATA/FEATURE SPACE.

Notation	Dimensions	Description
$\tilde{\mathbf{S}}_a^i$	$\mathbb{R}^{(T\times 320)\times 1}$	noisy sequence of T audio frames
$\mathbf{S}_v^i$	$\mathbb{R}^{T\times 90\times 110\times 3}$	sequence of T video frames
$\mathbf{Z}_a$	$\mathbb{R}^{T\times 512}$	embedding of audio sequence
$\mathbf{Z}_v$	$\mathbb{R}^{T\times 512}$	embedding of video sequence
$\mathbf{Z}_{av}$	$\mathbb{R}^{T\times 1024}$	joint embedding produced by MCB
$\mathbf{Y}_{av}$	$\mathbb{R}^{1\times 1024}$	the last temporal output of the last LSTM layer
$\mathbf{Out}_{av}$	$\mathbb{R}^{1\times 1}$	the final network's output representing speech presence/absence

IV. EXPERIMENTAL RESULTS

A. Training Process

1) Unimodal Training: Prior to training our deep multimodal end-to-end architecture, we train two unimodal variants of our architecture, for audio and video. This allows us

to initialize our multimodal network with weights from the learned unimodal networks, which enables the network to converge to a better minima while also improving convergence speed. To construct the audio unimodal network, we remove the MCB module from our multimodal architecture and merely feed the audio embedding $\mathbf{Z}_a$ directly to the classification block as described in Section III-D. Similarly, we construct the video network by removing the MCB and feeding the video embedding $\mathbf{Z}_v$ directly to the classification block.

For the training of the visual network, we initialize all the weights with values from a random normal distribution with zero mean and variance 0.01, including the weights of the ResNet-18 model. In our experiments, we found this leads to better results than initializing the ResNet-18 from a model that was pre-trained on, e.g., ImageNet [11]. We feed the network with sequences of $T = 15$ video frames, and they are all processed in parallel as discussed in Section III-A. The produced embedding, $\mathbf{Z}_v \in \mathbb{R}^{15\times512}$, is fed directly to the classification layers.

Similarly to the visual network, we initialize all the weights of the audio network with values from a random normal distribution with zero mean and variance 0.01. We feed the WaveNet encoder with sequences $\tilde{\mathbf{S}}_a^i$ of $T = 15$ raw audio frames, which corresponds to 4800 audio samples or 0.6 seconds of audio. The WaveNet encoder produces the embedding $\mathbf{Z}_a \in \mathbb{R}^{15\times512}$. As in the visual network, we feed $\mathbf{Z}_a$ directly to the classification layers.

We train each network separately for 150 epochs using stochastic gradient descent (SGD) with weight decay of 10^{-4} and momentum 0.9. We use an initial learning rate of 0.01 and divide this learning rate by a factor of 10 every 30 epochs. For the audio network we use a mini-batch of 96 samples and for the visual network a mini-batch of size 16. As described in Section III-D, we use dropout on the outputs of the WaveNet encoder and the ResNet-18 model and also on the last fully connected layer for regularization of the network.

2) Multimodal Training: Once both unimodal networks are trained, the classification block of each unimodal network is discarded, and we use the learned weights of the WaveNet encoder and ResNet-18 modules to initialize the corresponding parts of the multimodal network. We use MCB with an output size of 1024 to fuse the feature vectors extracted from both modalities. This joint representation is fed to a similar classification block used in the unimodal variants. The LSTM and fully connected layers in the classification block are initialized with random weights from a random normal distribution with zero mean and variance 0.01. The entire network is trained in an end-to-end manner, and the visual and speech networks are fine-tuned. The multimodal network is trained for an additional 50 epochs using SGD with weight decay of 10^{-4}, momentum 0.9 and a fixed learning rate of 0.001.

For further regularization of the network, and in order to avoid the exploding gradients problem which can arise in LSTM cells, we enforce a hard constraint on the norm of the gradients by scaling it when it exceeds a threshold of 0.5 [54].

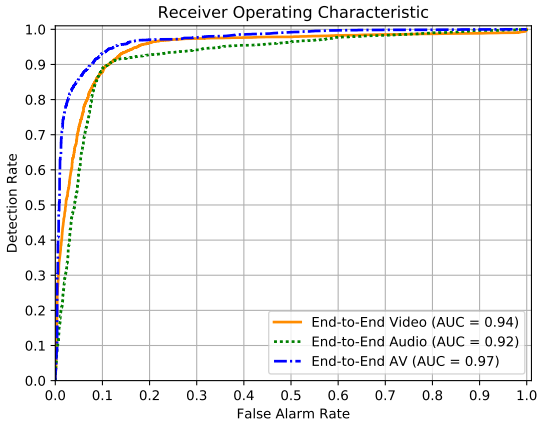

Fig. 3. Probability of detection versus probability of false alarm of our unimodal and multimodal architectures (best viewed in color).

B. Evaluation

In order to demonstrate the benefit of the fusion of the audio and video signals for voice activity detection, we evaluated both the multimodal and the unimodal versions of the proposed architecture. The unimodal versions are denoted in the plots by "End-to-End Audio" and "End-to-End Video", respectively, and the multimodal version is denoted in the plots by "End-to-End AV". The unimodal and multimodal versions are constructed and trained as described in Section IV-A. The benefit of fusing the audio and the video modalities is clearly shown in Fig. 3, where the proposed audio-visual architecture significantly outperforms the unimodal versions. We compare the different networks in the form of receiver operating characteristic (ROC) curves, which present the probability of detection versus the probability of false alarm. We also give the area under the curve (AUC) measure for each of the architectures.

To evaluate the performance of the proposed deep multimodal end-to-end architecture, we compare it with the competing audio-visual voice activity detectors presented in [34] and [36]. We evaluated all voice activity detectors on the challenging evaluation set described in Section II-B. In [34], the authors used only the four largest components of their diffusion mapping to describe the audio/video signals. In our experiments, we found that using a larger number of components to describe the signals, can be beneficial. We experimented with a different number of diffusion mapping components between 4 and 20 and found that the performance improved up to the level of using ten components, and increasing the number of components even further did not provide any additional noticeable improvement in performance. This can be explained by the more complex nature of our evaluation set, in which each frame is contaminated with a different combination of background noise, transient interference, and SNR level. We denote the VAD proposed in [34] using 4,6 and 10 components as "Dov4 AV", "Dov6 AV" and "Dov10 AV" respectively. The VAD proposed in our earlier work [36] was

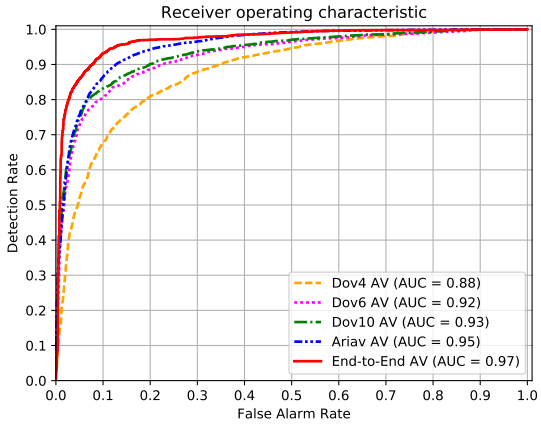

Fig. 4. Probability of detection versus probability of false alarm of our multimodal end-to-end architecture and the VADs presented in [34] and [36] (best viewed in color).

found to be superior to all versions of the VADs presented in [34] and is denoted in the plots by "Ariav AV". In Fig. 4 it is clearly shown that the performance of our proposed deep multimodal end-to-end architecture is superior to those presented in [34] and [36].

We perform several ablation experiments to demonstrate the effectiveness of our proposed end-to-end architecture. To demonstrate the advantages of fusing the embeddings $\mathbf{Z}_a$ and $\mathbf{Z}_v$ using the MCB module, we conducted an experiment in which we replaced the MCB module with a simple vector concatenation, which is standard practice today for multimodal problems. In another set of experiments, we construct a multimodal network from the two unimodal networks in which we do not remove the unimodal LSTM layers. Instead, we feed the MCB module with the outputs of the two unimodal LSTMs and the joint representation produced by the MCB is fed directly to a fully connected layer for classification. We denote this architecture as "separate-LSTM", as opposed to the originally proposed architecture which we denote in the table as "shared-LSTM". Throughout all of the above experiments, the rest of the architecture, including the number of neurons/cells in each layer and the training procedure remains unchanged. Moreover, since the MCB's output size was chosen to be 1024, it matches the size of the concatenated embedding, so it is a fair comparison between the two variants. Table II shows the results in terms of classification accuracy, precision, recall and f1-score on the evaluation set. It can be seen that fusing the two modalities using MCB gives better results than a simple concatenation of the feature vectors. Moreover, the architecture with the shared-LSTM shows better performance, and a possible explanation is that this way the LSTM can capture the dynamics of the speech signal which is shared across the modalities.

Furthermore, we experimented with different sequence lengths T to feed our end-to-end network. We conducted experiments using sequence lengths of $T = \{15, 30, 45\}$,

TABLE II
ACCURACY, PRECISION, RECALL AND F1-SCORE ON EVALUATION SET FOR DIFFERENT END-TO-END MULTIMODAL ARCHITECTURES. WE DENOTE THE PROPOSED ARCHITECTURE AS "SHARED LSTM" AND THE ARCHITECTURE IN WHICH WE USE UNIMODAL LSTMS AS "SEPARATE LSTM"

Architecture	Acc.	Precision	Recall	F1 Score
separate-LSTM+concat	0.8962	0.8759	0.9205	0.8977
separate-LSTM+MCB	0.9084	**0.9125**	0.8836	0.8978
shared-LSTM+concat	0.8982	0.8705	**0.9356**	0.9018
shared-LSTM+MCB	**0.9152**	0.9033	0.9254	**0.9142**

which correspond to 0.6, 1.2 and 1.8 seconds of audio/video. In these experiments, we used our multimodal networks with MCB architecture. Note that the change of T merely affects the number of sequences in the training and evaluation sets in a negligible way, so it is a fair comparison between the different cases. Table III shows the results in terms of classification accuracy on the evaluation set. As seen in Table III, there is no real advantage to using longer sequences as input, but the computational cost and memory consumption are greater, mainly due to the vision network, and therefore we opted to use a sequence length of $T = 15$.

TABLE III
ACCURACY ON EVALUATION SET FOR DIFFERENT SEQUENCE LENGTHS T FED INTO OUR END-TO-END MULTIMODAL ARCHITECTURE

Sequence length	Acc.
15	0.9152
30	0.9152
45	0.9148

In contrast to our previous work [36], where the network operates on hand-crafted features extracted from the two modalities, the proposed architecture operates on raw signals to extract the most suitable features from each modality to the task of voice activity detection. Moreover, in [36] the features from the two modalities are merely concatenated before being fed to an autoencoder to exploit the relations between the modalities. However, in our proposed architecture the fusion is performed via an MCB module which allows for higher order relations between the two modalities to be exploited.

V. CONCLUSIONS AND FUTURE WORK

We have proposed a deep multimodal end-to-end architecture for speech detection, which utilizes residual connections and dilated convolutions and operates on raw audio and video signals to extract meaningful features in which the effect of transients is reduced. The features from each modality are fused into a joint representation using MCB which allows higher order relations between the two modalities to be explored. In order to further exploit the differences in the dynamics between speech and the transients, the joint representation is fed into a deep LSTM. A fully connected layer is added, and the entire network is trained in a supervised manner to perform voice activity detection. Experimental

results have demonstrated that our multimodal end-to-end architecture outperforms unimodal variants while providing accurate detections even under low SNR conditions and in the presence of challenging types of transients. Furthermore, the use of MCB for modality fusion has also been shown to outperform other methods for modality fusion.

Future research directions include applying the proposed end-to-end multimodal architecture to different voice-related tasks such as speech recognition or speech enhancement. Moreover, we will explore additional methods for noise injection, in which the video signal is augmented and the speakers' voices are modified according to the injected noise levels (Lombard effect).

Another direction worth exploring would be to use the proposed architecture on altogether different modalities, e.g., replacing the audio signal with an electrocardiogram (ECG) signal and training the network to perform ECG related tasks. This is possible since the architecture operates on raw signals and therefore does not depend on audio, or image, specific features.

ACKNOWLEDGEMENTS

The authors thank the associate editor and the anonymous reviewers for their constructive comments and useful suggestions.

REFERENCES

[1] David A. Krubsack and Russel J. Niederjohn, "An autocorrelation pitch detector and voicing decision with confidence measures developed for noise-corrupted speech," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 319–329, 1991.

[2] J.-C. Junqua, Brian Mak, and Ben Reaves, "A robust algorithm for word boundary detection in the presence of noise," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 3, pp. 406–412, 1994.

[3] Stefaan Van Gerven and Fei Xie, "A comparative study of speech detection methods," in *Proc. 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 1095 – 1098, 1997.

[4] Namgook Cho and Eun-Kyoung Kim, "Enhanced voice activity detection using acoustic event detection and classification," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 196–202, 2011.

[5] Joon-Hyuk Chang, Nam Soo Kim, and Sanjit K. Mitra, "Voice activity detection based on multiple statistical models," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 1965–1976, 2006.

[6] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.

[7] Javier Ramírez, José C. Segura, Carmen Benítez, Angel De La Torre, and Antonio Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, no. 3, pp. 271–287, 2004.

[8] David Dov, Ronen Talmon, and Israel Cohen, "Kernel method for voice activity detection in the presence of transients," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2313–2326, 2016.

[9] Jong Won Shin, Joon-Hyuk Chang, and Nam Soo Kim, "Voice activity detection based on statistical models and machine learning approaches," *Computer Speech & Language*, vol. 24, no. 3, pp. 515–530, 2010.

[10] Ji Wu and Xiao-Lei Zhang, "Maximum margin clustering based statistical VAD with multiple observation compound feature," *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 283–286, 2011.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. CVPR09*, 2009.

[12] John S Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993.

[13] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh, "VQA: Visual Question Answering," in *Proc. International Conference on Computer Vision (ICCV)*, 2015.

[14] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[15] Xiao-Lei Zhang and Ji Wu, "Deep belief networks based voice activity detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697–710, 2013.

[16] Samuel Thomas, Sriram Ganapathy, George Saon, and Hagen Soltau, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2519–2523.

[17] Rasool Tahmasbi and Sadegh Rezaei, "A soft voice activity detection using garch filter and variance gamma distribution," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1129–1134, 2007.

[18] Simon Leglaive, Romain Hennequin, and Roland Badeau, "Singing voice detection with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 121–125.

[19] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6645–6649.

[20] Wei-Tyng Hong and Chien-Cheng Lee, "Voice activity detection based on noise-immunity recurrent neural networks," *International Journal of Advancements in Computing Technology (IJACT)*, vol. 5, no. 5, pp. 338–345, 2013.

[21] Thad Hughes and Keir Mierle, "Recurrent neural networks for voice activity detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7378–7382.

[22] Wootaek Lim, Daeyoung Jang, and Taejin Lee, "Speech emotion recognition using convolutional and recurrent neural networks," in *Proc. Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–4.

[23] David Sodoyer, Bertrand Rivet, Laurent Girin, J.-L. Schwartz, and Christian Jutten, "An analysis of visual speech information applied to voice activity detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006, vol. 1, pp. 601–604.

[24] David Sodoyer, Bertrand Rivet, Laurent Girin, Christophe Savariaux, Jean-Luc Schwartz, and Christian Jutten, "A study of lip movements during spontaneous dialog and its application to voice activity detection," *Journal of the Acoustical Society of America*, vol. 125, no. 2, pp. 1184–1196, 2009.

[25] Andrew Aubrey, Bertrand Rivet, Yulia Hicks, Laurent Girin, Jonathon Chambers, and Christian Jutten, "Two novel visual voice activity detectors based on appearance models and retinal filtering," in *Proc. 15th European Signal Processing Conference (EUSIPCO)*, 2007, pp. 2409–2413.

[26] Qingju Liu, Wenwu Wang, and Philip Jackson, "A visual voice activity detection method with adaboosting," *Proc. Sensor Signal Processing for Defence (SSPD)*, pp. 1–5, 2011.

[27] AJ Aubrey, YA Hicks, and JA Chambers, "Visual voice activity detection with optical flow," *IET Image Processing*, vol. 4, no. 6, pp. 463–472, 2010.

[28] P. Tiawongsombat, Mun-Ho Jeong, Joo-Seop Yun, Bum-Jae You, and Sang-Rok Oh, "Robust visual speakingness detection using bi-level HMM," *Pattern Recognition*, vol. 45, no. 2, pp. 783–793, 2012.

[29] Ibrahim Almajai and Ben Milner, "Using audio-visual features for robust voice activity detection in clean and noisy speech," in *Proc. 16th European Signal Processing Conference (EUSIPCO)*, 2008.

[30] Takami Yoshida, Kazuhiro Nakadai, and Hiroshi G. Okuno, "An improvement in audio-visual voice activity detection for automatic speech recognition," in *Proc. 23rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2010, pp. 51–61.

[31] Vicente P. Minotto, Carlos BO Lopes, Jacob Scharcanski, Claudio R. Jung, and Bowon Lee, "Audiovisual voice activity detection based on microphone arrays and color information," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 147–156, 2013.

[32] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng, "Multimodal deep learning," in *Proc. 28th International Conference on Machine Learning (ICML)*, 2011, pp. 689–696.

[33] Shiqing Zhang, Shiliang Zhang, Tiejun Huang, and Wen Gao, "Multimodal deep convolutional neural network for audio-visual emotion recognition," in *Proc. ACM on International Conference on Multimedia Retrieval (ICMR'16)*. ACM, 2016, pp. 281–284.

[34] David Dov, Ronen Talmon, and Israel Cohen, "Audio-visual voice activity detection using diffusion maps," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 732–745, 2015.

[35] Ronald R. Coifman and Stéphane Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[36] Ido Ariav, David Dov, and Israel Cohen, "A deep architecture for audio-visual voice activity detection in the presence of transients," *Signal Processing*, vol. 142, pp. 69–74, 2018.

[37] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. International Conference on Machine Learning (ICML)*, pp. 1764–1772.

[38] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A. Nicolaou, Björn Schuller, and Stefanos Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5200–5204.

[39] Panagiotis Tzirakis, George Trigeorgis, Mihalis A Nicolaou, Björn W Schuller, and Stefanos Zafeiriou, "End-to-end multimodal emotion recognition using deep neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1301–1309, 2017.

[40] Jen-Cheng Hou, Syu-Siang Wang, Ying-Hui Lai, Yu Tsao, Hsiu-Wen Chang, and Hsin-Min Wang, "Audio-visual speech enhancement using multimodal deep convolutional neural networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 117–128, 2018.

[41] Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic, "End-to-end audiovisual speech recognition," *arXiv preprint arXiv:1802.06424*, 2018.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

[43] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio.," in *Proc. SSW*, p. 125.

[44] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell, "Compact bilinear pooling," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 317–326.

[45] *[Online]. Available: http://www.freesound.org*.

[46] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," 2016.

[47] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th International Conference on Machine Learning (ICML)*, pp. 807–814.

[48] Joshua B. Tenenbaum and William T. Freeman, "Separating style and content with bilinear models," *Neural Computation*, vol. 12, no. 6, pp. 1247–1283, 2000.

[49] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1449–1457.

[50] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *arXiv preprint arXiv:1606.01847*, 2016.

[51] Moses Charikar, Kevin Chen, and Martin Farach-Colton, "Finding frequent items in data streams," in *Proc. International Colloquium on Automata, Languages, and Programming*. Springer, 2002, pp. 693–703.

[52] Ninh Pham and Rasmus Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in *Proc. 19th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2013, pp. 239–247.

[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[54] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *Proc. International Conference on Machine Learning (ICML)*, pp. 1310–1318.