# 3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition

Amirsina Torfi, Seyed Mehdi Iranmanesh, Nasser Nasrabadi, *Fellow, IEEE* and Jeremy Dawson

***Abstract*—Audio-visual recognition (AVR) has been considered as a solution for speech recognition tasks when the audio is corrupted, as well as a visual recognition method used for speaker verification in multi-speaker scenarios. The approach of AVR systems is to leverage the extracted information from one modality to improve the recognition ability of the other modality by complementing the missing information. The essential problem is to find the correspondence between the audio and visual streams, which is the goal of this work. We propose the use of a coupled 3D Convolutional Neural Network (3D-CNN) architecture that can map both modalities into a representation space to evaluate the correspondence of audio-visual streams using the learned multimodal features. The proposed architecture will incorporate both spatial and temporal information jointly to effectively find the correlation between temporal information for different modalities. By using a relatively small network architecture and much smaller dataset for training, our proposed method surpasses the performance of the existing similar methods for audio-visual matching which use 3D CNNs for feature representation. We also demonstrate that an effective pair selection method can significantly increase the performance. The proposed method achieves relative improvements over 20% on the Equal Error Rate (EER) and over 7% on the Average Precision (AP) in comparison to the state-of-the-art method.**

***Index Terms*—Convolutional Networks, 3D Architecture, Deep Learning, Audio-visual Recognition.**

## I. Introduction

THE crucial part of an AVR algorithm is the feature selection for both audio and visual modalities, which has a direct impact on the performance of the audio-visual recognition task. Regarding the speech modality, most speech recognition systems employ Hidden Markov Models (HMMs) to extract the temporal information of speech and Gaussian Mixture Models (GMMs) to discriminate between different HMMs states for acoustic input representation. However deep learning has recently been employed as a mechanism for unsupervised speech feature extraction [1]. Beyond speaker and speech recognition, deep learning has also been used for feature extraction of unlabeled facial images [2]. Similar approaches have been employed in the analysis of multi-modal voice and face data, which resulted in an improvement of speech recognition performance [3].

The inference based on common sense is that the lip motions and the heard voice which is represented by speech features are highly correlated as a human is usually able to match the heard sound to a given set of lip motion. However, the visual lip motions and their corresponding audio stream still can have non-negligible uncorrelated information. Decision fusion has been shown to be effective in which the final decision is made by fusing the statistically independent decisions from different modalities with the emphasize on uncorrelated characteristics between different modalities [4]. However, data fusion in early stages, demonstrated more promising results as it creates a joint representation between two modalities based on the cross-modality correlations [5], [6]. As the corresponding audio-visual streams have correlated and uncorrelated information, we propose an architecture based on Deep Neural Networks (DNNs) as a discriminative model between the two modalities in order to simultaneously distinguish between the correlated and uncorrelated components.

Alongside with the audio stream, lip motions can also contain speaker-related information. Some research efforts applied both modalities for Speaker Identification (SI) and Speaker Verification (SV) mainly based on decision fusion and MFCC features [7], [8]. The speaker dependent systems are generally aimed to recognize the speech or speaker identity based on speaker-dependent characteristics. However, speaker-independent systems must be able to recognize the part of speech regardless of who is speaking. The SV has two general categories of text-dependent and text-independent types. In text-dependent setup, a fixed text is used for all experiments. On the other hand, in text-independent SV, no prior information or restrictions are considered for the utterances. It makes the text-independent to be more challenging that text-dependent scenario. Most of the previous research efforts for audio-visual for the aforementioned problems have been conducted in the text-dependent scenario. In contrast, we conduct our experiments in speaker-independent and text-independent mode to deliberately investigate the most challenging scenario.

There is a significant amount of literature describing audio-visual recognition in a variety of applications, including speech recognition in noisy environments using lip motions as auxiliary features [9], [10], as well as the converse where speech data is leveraged for the purpose of lip reading [11], [12]. However, there is a lack of research on concurrently incorporating the spatial and temporal audio-visual information to address the root problem of whether or not the audio stream and the visual stream match. As an example, in a multi-speaker scenario, if the features connecting audio and video can be found, the speakers' lip motions could be determined by the audio stream and vice versa. In this paper, we investigate the main problem of audio-visual matching. In another word, the main problem is to recognize whether the visual lip motions of a speaker corresponds to the accompanying speech signal. The aforementioned root problem is the precedent to audio-visual synchrony verification, as recognizing the consistency between the audio-visual streams is desired. The problem of audio-visual synchrony recognition has been addressed in

different research efforts such as [13] for identity verification, and liveness recognition of the audio-visual streams [14], [15].

To address the problem, we propose to use the 3D Convolutional Neural Networks models that have recently been employed for action recognition, scene understanding, and speaker verification and demonstrated promising results [16]–[18]. 3D CNNs concurrently extract features from both spatial and temporal dimensions, so the motion information is captured and concatenated in adjacent frames. We use 3D CNNs to generate separate channels of information from the input frames. The combination of all channels of correlated information creates the final feature representation.

The focus of the research effort described in this paper is to implement two non-identical 3D-CNNs for audio-visual matching (Section V). The goal is to design nonlinear mappings that learn a non-linear embedding space between the corresponding audio-video streams using a simple distance metric. This architecture can be learned by evaluating pairs of audio-video data and later used for distinguishing between pairs of matched and non-matched audio-visual streams. One of the main advantages of our audio-visual model is the noise-robust audio features, which are extracted from speech features with a locality characteristic (Section IV), and the visual features, which are extracted from spatial and temporal information of lip motions. Both audio-visual features are extracted using 3D CNNs, allowing the temporal information to be treated separately for better decision making.

The contributions of this paper are as follows:

- A novel coupled 3D CNN architecture, which simultaneously extracts spatial and temporal information, is designed with a significant reduction in dimension compared to the input space and is optimized for distinguishing between match and non-match audio-visual streams.
- The network is relatively small, which has the advantage of allowing it to be easily trainable and fast to test.
- Compared to traditional MFCCs, a different type of speech feature has been used for representing the audio stream, which provides more promising results.
- An adaptive online pair selection method with the output feature space distance as a criterion has been proposed for selecting the main contributing pairs for accelerating the convergence speed and preventing over-fitting.

To the best of our knowledge, this is the first attempt to use 3D convolutional neural networks for audio-visual matching in which a bridge between spatiotemporal features has been established to build a common feature space between audio-visual modalities. The source code[1] of this paper has been released online as an open source project [19].

## II. Related works

Lip reading and audio-visual speech recognition (AVSR) are highly correlated such that the relevant information of one modality can improve the recognition of the other modality in any of the two aforementioned applications. DNNs have been employed for fusing speech and visual modalities for audio-visual automatic speech recognition (AV-ASR) [20]. Moreover, in [21], a connectionist HMM system is introduced for AVSR and a pre-trained CNN is used to classify phonemes.

Some researchers have used CNNs to predict and generate phonemes [22] or visemes [23] without considering the word-level prediction. Phonemes are the smallest distinguishable unit of an audio stream which are combined to create a spoken word, and a viseme is its corresponding visual equivalent. For recognizing full words, a Long Short-Term Memory (LSTM) classifier with Discrete Cosine Transform (DCT) and Deep Bottleneck Features (DBF) is trained [24]. Similarly, LSTM with Histogram of Oriented Gradients (HOG) features are used also for phrase recognition [25].

One of the most challenging applications of audio-visual recognition is the audio-video synchronization for which the audio-visual matching ability is required. The research efforts related to this paper consist of different audio-visual recognition tasks which try to find the correspondence between the two modalities. Different approaches have been employed for tackling the audio-visual matching problem. Some are based on data-driven approaches, such as using DNN classifiers to determine the off-sync time [26], [27] and some are based on Canonical Correlation Analysis (CCA) [28] and Co-Inertia Analysis (CoIA).

The most relevant work to ours is the work of Chung and Zisserman [26] which is aimed at determining the audio-video synchronization between lip motions and an audio stream in a video. They use traditional Mel-frequency cepstral coefficients (MFCCs) to present speech features. In [26], CNNs have been applied for feature construction with the motion information as the input depth, which does not effectively reflect the correlation and distinction between spatial and temporal information. Instead, we propose to apply 3D convolutional layers to simultaneously capture the spatial and temporal discriminative features independently. Moreover, we designed an adaptive pair selection method for improving the accuracy and accelerating the training convergence, as opposed to using the whole training data which has been used in [26].

## III. Dataset

The datasets that have been used for our experiments are the *Lip Reading in the Wild (LRW)* [29] and the *WVU Audio-Visual Dataset* dataset (*AVD*) [30]. The LRW dataset consists of up to 1000 utterances of 500 different words, spoken by different speakers. All videos are 1.16 seconds in length, and the word occurs in the middle of the video. The *AVD* dataset consists of audio and video data collected over a period spanning 2014 through 2015. The video and audio data consists of both scripted and unscripted voice samples. For the scripted samples, the participant read a sample of text. For the unscripted samples, the participant answered interview questions that prompted conversational responses rather than simple 'yes' or 'no' answers.

### A. Processing

The processing pipeline of both datasets is shown in Fig. 1. The pipeline is subdivided into two visual and audio sections. In the visual section, the videos are post-processed

[1] https://github.com/astorfi/lip-reading-deeplearning

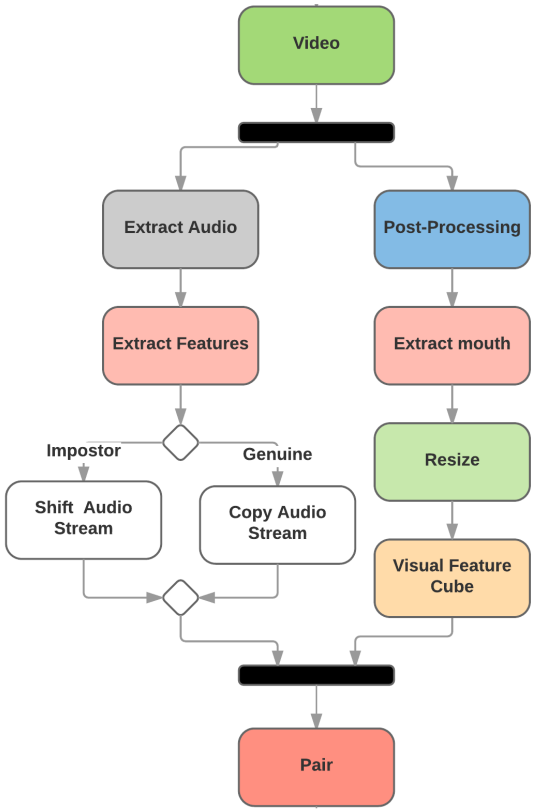


Fig. 1. The processing pipeline of the datasets.

to have equal frame rate of 30 f/s. Then, face tracking and mouth area extraction is performed on the videos using the dlib library [31]. Finally, all mouth areas are resized to have the same size, and concatenated to form the input feature cube. The dataset does not contain any audio files. In the audio section, the audio files are extracted from videos using the FFmpeg framework [32]. Then the speech features will be extracted from audio files. The library that has been used for speech feature extraction task is SpeechPy [33].

## IV. Data Representation

The proposed architecture utilizes two non-identical ConvNets which uses a pair of speech and video streams. The network input is a pair of features that represent lip movement and speech features extracted from 0.3-second of a video clip. The main task is to determine if a stream of audio corresponds with a lip motion clip within the desired stream duration.

The difficulty of this task is the short time interval of the video clip (0.3-0.5 second) considered to evaluate the method. This setting is close to real-world scenarios because, in some biometrics or forensics applications, only a short amount of captured video or audio might be available to distinguish between different modalities. Temporal video and audio features must correspond over the time interval they cover. This correspondence is discussed in the next two sections.

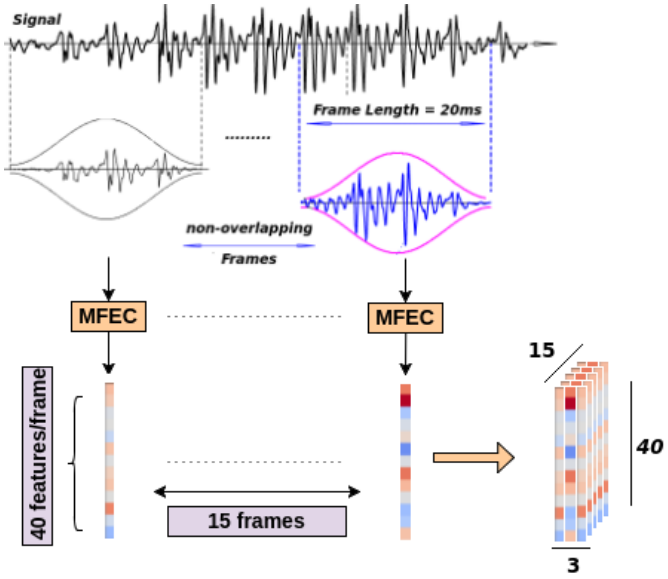


Fig. 2. Speech feature generation using stacked frames created from the input signal samples.

### A. Speech

The main characteristic of CNNs is their locality, i.e., the convolution operation is applied to specific local regions in an image. As a visual inference of this locality property, the neighbor features should be correlated in some sense. Since the input speech feature maps are treated as images when a CNN architecture is used, the features must be locally correlated in the sense of time and frequency on both axes respectively.

The MFCCs, which are derived from the cepstral representation of the audio stream, can be used as the speech feature representation. However, the drawback is that they have non-local characteristics. The reason for this is that during the last operation (DCT[2]) for generating MFCCs, which is aimed at eliminating the correlations between energy coefficients, the order of the filter-bank energies is changed, which leads to disturbing the locality property. The approach employed in this paper is to use the log-energies derived directly from the filter-banks energies which we call MFECs[3]. The extraction of MFECs is similar to MFCCs, but with no DCT operation.

On the time axis, the temporal features are non-overlapping 20ms windows which are used for the generation of spectrum features that possess a local characteristic. The input speech feature map, which is represented as an image cube, corresponds to the spectrogram, as well as the first and second order derivatives of the MFEC features. These three channels correspond to the image depth. This representation is depicted in Fig. 2. Collectively, from a 0.3-second clip, 15 temporal feature sets (each forms a 40 MFEC features) can be derived which form a speech feature cube. Each input feature map for a single audio stream has a dimensionality of $15 \times 40 \times 3$.

Similar approaches are used in [34] for automatic speech recognition (ASR), where local filtering layers are employed

[2]Discrete Cosine Transform
[3]Mel-frequency energy coefficients

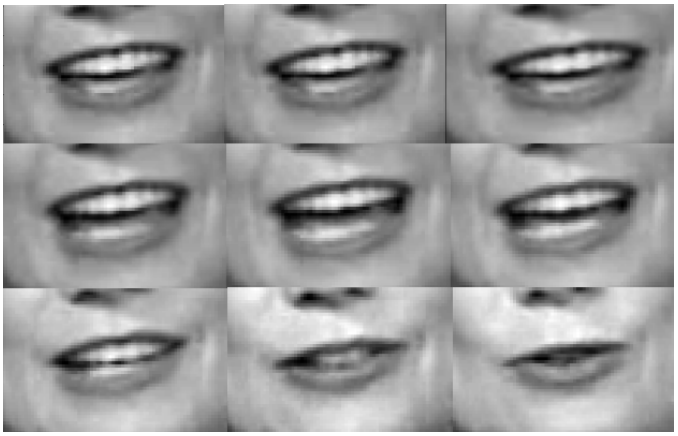

Fig. 3. The sequence of mouth areas in a 0.3-second video stream.

to extract and represent the spatial speaker-independent features. Similar input features are employed in [26]. However, these efforts used MFCC features as the speech feature representations without the use of the temporal derivatives.

### B. Video

The frame rate of each video clip used in this effort is 30 f/s. Consequently, 9 successive image frames form the 0.3-second visual stream. The input of the visual stream of the network is a cube of size 9x60x100, where 9 is the number of frames that represent the temporal information. Each channel is a 60x100 gray-scale image of mouth region. An example of mouth area representation is provided in Fig. 3.

A relatively small mouth crop region is intentionally chosen due to practical consideration because, in real-world scenarios, it is not very likely to have access to high-resolution images. Moreover, unlike the usual experimental setups for CNNs, we did not restrict our experiments to input images with uniformly square aspect ratios.

### C. Preprocessing

Neither mean-feature-normalization[4] nor Min-Max scaling[5] demonstrated promising results in the testing phase for our experiments. However, both improved the training optimization. We found the data standardization ($\frac{X-\bar{X}}{\sigma}$) to be effective as a preprocessing operation.

## V. Architecture

The architecture is a coupled 3D convolutional neural network in which two different networks with different sets of weights must be trained. For the visual network, the lip motions' spatial and temporal information are incorporated jointly and will be fused for exploiting the temporal correlation. For the audio network, the extracted energy features are considered as a spatial dimension, and the stacked audio frames create the temporal dimension. In our proposed

[4] $\frac{X-\bar{X}}{X_{max}-X_{min}}$

[5] $\frac{X-X_{min}}{X_{max}-X_{min}}$

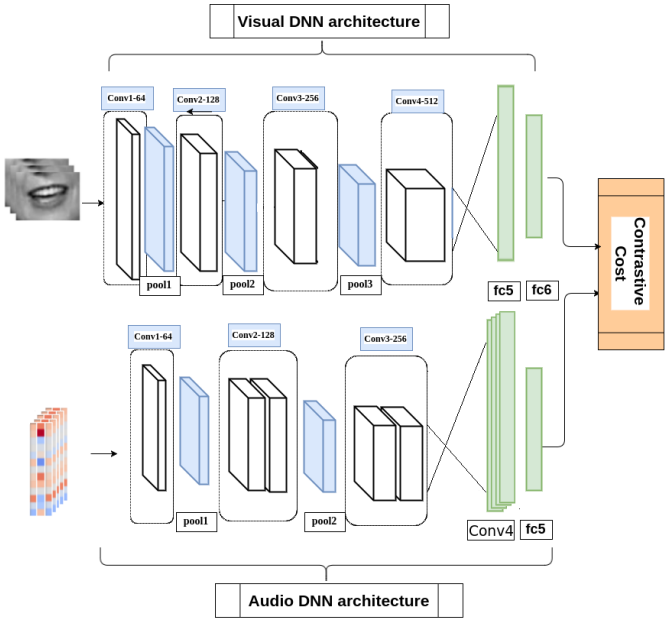


Fig. 4. **Outline of the architecture.** Two non-identical 3D CNNs with audio and visual streams as inputs are coupled together in the last fully-connected layer by contrastive cost criterion.

3D CNN architecture, the convolutional operations are performed on successive temporal frames for both audio-visual streams. Dropout($\rho$) has been used for all fully-connected layers prior to the last layer. Except for the last layer, all layers are followed by PReLU activation, as proposed in [35] which is a generalization to ReLU. Compared to ReLU activation, PReLU employment demonstrated better performance in our experiments in which the parameters for rectifiers are learned adaptively. The architecture is depicted in Fig. 4.

### A. Visual network

The network architecture used for training on video streams is described in Table I. In Table I, the spatial size of the 3D kernels is reported as $T \times H \times W$ where $T$ is the kernel size in temporal dimension, and $H$ and $W$ are the kernel sizes in height and width dimensions, respectively. As with usual CNN architectures, the kernel depth is equal to the input-channel size, which is the number of the feature maps of the previous layer. For kernel spatial size representation, the kernel depth sizes are discarded for simplicity.

An important characteristic of the visual network is its pooling method. Since we are using 3D convolutional layers, 3D pooling layers are also utilized. Although 1x3x3 kernels are applied for spatial feature pooling, in order to increase robustness to the moving lip effect[6], the pooling stride is set to two[7] in order to maintain lip movement features[8] in the neighborhood of the pooling kernel. The 3D convolutional operations are performed to find the correlation between high-level temporal and spatial information by fusion among them. No zero-padding is used in the visual architecture.

[6] The lip place in the video clip is not necessarily fixed

[7] It means a kind of overlap because the pooling kernel is 1x3x3

[8] More importantly high-level features

TABLE I
THE ARCHITECTURE FOR VIDEO STREAM.

| layer | input-size | output-size | kernel | stride |
|---|---|---|---|---|
| Conv1 | 9x60x100x1 | 7x58x98x16 | 3x3x3 | 1 |
| Pool1 | 7x58x98x16 | 7x28x48x16 | 1x3x3 | 1x2x2 |
| Conv2 | 7x28x48x16 | 5x26x46x32 | 3x3x3 | 1 |
| Pool2 | 5x26x46x32 | 5x12x22x32 | 1x3x3 | 1x2x2 |
| Conv3 | 5x12x22x32 | 3x10x20x64 | 3x3x3 | 1 |
| Pool3 | 3x10x20x64 | 3x4x9x64 | 1x3x3 | 1x2x2 |
| Conv4 | 3x4x9x64 | 1x2x7x128 | 3x3x3 | 1 |
| FC5 | 1x2x7x128 | 256 | - | - |
| FC6 | 256 | 64 | - | - |

### B. Audio network

The network architecture used for training on audio streams is described in Table II. In our architecture, pooling operations are only applied in the frequency axis (domain) to maintain the temporal information within the time frames. Additionally, our proposed architecture has a high level of compression in which only 64 output units are used.

As with Table I, in Table II, the spatial size of the 3D kernels is reported as $T \times H \times W$, and the kernel depth sizes are discarded for simplicity as well. A 3D kernel is used in the first layer for spatiotemporal feature extraction using a 3D convolutional operation. Except for the first layer, since the spatial dimension for the audio feature maps are $M \times N \times 1$ in higher-level layers, we are, in essence, dealing with 2D dimensionality. Because of this, we have regular 2D convolutional operations for the audio network which simultaneously capture the temporal and spatial information using their 2D kernels. The kernel spatial dimensions are deliberately demonstrated as $T \times H \times 1$ to emphasize the connection between the 3D and 2D convolutional operations.

TABLE II
THE ARCHITECTURE FOR AUDIO STREAM.

| layer | input-size | output-size | kernel | stride |
|---|---|---|---|---|
| Conv1 | 15x40x3 | 13x36x1x16 | 3x5x3 | 1 |
| Pool1 | 13x36x1x16 | 13x18x1x16 | 1x2x1 | 1x2x1 |
| Conv2-1 | 13x18x1x16 | 11x15x1x32 | 3x4x1 | 1 |
| Conv2-2 | 11x15x1x32 | 9x12x1x32 | 3x4x1 | 1 |
| Pool2 | 9x12x1x32 | 9x6x1x32 | 1x2x1 | 1x2x1 |
| Conv3-1 | 9x6x1x32 | 7x4x1x64 | 3x3x1 | 1 |
| Conv3-2 | 7x4x1x64 | 5x2x1x64 | 3x3x1 | 1 |
| Conv4 | 5x2x1x64 | 3x1x1x128 | 3x2x1 | 1 |
| FC5 | 3x1x1x128 | 64 | - | - |

In the audio architecture, zero-padding is not used because zero-padding adds extra virtual zero-energy coefficients which are meaningless in the sense of local feature extraction. Another important characteristic is the use of non-square kernels. As we go from low-level features (lower level convolutional layers) to high-level features (higher level convolutional layers), the kernel widths follow a decreasing order. This setup results in extraction and processing of more temporal features in the lower level that are related to speech features, as well as and correlated features in the high-level features that are the features extracted from the CNN.

### C. Coupling

Both audio-visual networks are coupled at their highest level, which is the last fully-connected layer with a cardinality of $\zeta$. The $\zeta$ parameter represents the cardinality of the output embedding. The default value for $\zeta$ is 64, which forms our base experiments. Since the network is trained in verification mode, in order to match the audio and visual representation of the spoken parts of speech, a contrastive loss is used as a discriminative distance metric to optimize the coupling process, which has been proposed in [36]. The contrastive loss function $L_W(Y, X)$ is as follows:

$$L_W(Y, X) = \frac{1}{N}\sum_{i=1}^{N} L_W(Y_i, (X_{p_1}, X_{p_2})_i), \tag{1}$$

where N is the number of training samples, $(X_{p_1}, X_{p_2})_i$ is the i-th input pair, $Y_i$ is the corresponding label and $L_W(Y_i, (X_{p_1}, X_{p_2})_i)$ is defined as follows:

$$\begin{aligned} L_W(Y_i, (X_{p_1}, X_{p_2})_i) &= Y \times L_{gen}(D_W(X_{p_1}, X_{p_2})_i) \\ &+ (1 - Y) \times L_{imp}(D_W(X_{p_1}, X_{p_2})_i) + \lambda ||W||_2, \end{aligned} \tag{2}$$

in which $D_W(X_{p_1}, X_{p_2})$ is the Euclidean distance between the outputs of the network with $(X_{p_1}, X_{p_2})$ as the input. The last term is for regularization, and $\lambda$ is the regularization parameter. Finally, $L_{gen}$ and $L_{imp}$ are defined as the functions of $D_W(X_{p_1}, X_{p_2})$ by the following equations:

$$\begin{cases} L_{gen}(D_W(X_{p_1}, X_{p_2})) = \frac{1}{2} D_W(X_{p_1}, X_{p_2})^2 \\ L_{imp}(D_W(X_{p_1}, X_{p_2})) = \frac{1}{2} max\{0, (\mu - D_W(X_{p_1}, X_{p_2}))\}^2, \end{cases} \tag{3}$$

where $\mu$ is the predefined margin. Contrastive loss is employed as a mapping criteria, which will ideally place genuine pairs[9] to nearby and impostor pairs[10] to distant manifolds in the output space.

## VI. EVALUATION AND VERIFICATION METRIC

In this paper, we evaluate experimental results using the Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves characteristics. The ROC curve consists of the Validation Rate (VR) and False Acceptance Rate (FAR). All pairs $(X_{P_1}, X_{P_2})$ of the same identity are denoted with $\mathcal{P}_{gen}$ whereas all pairs belonging to different identities are denoted as $\mathcal{P}_{imp}$. If $D_W$ is the Euclidean distance between the outputs of the network with $(X_{P_1}, X_{P_2})$ as the input, we define true positive and false acceptance as below:

$$TP(\tau) = \{(X_{P_1}, X_{P_2}) \in \mathcal{P}_{gen}, D_W \leq \tau\}. \tag{4}$$

[9]Also called as match pair in which both samples in a pair belong to the same identity

[10]Also called as non-match pair in which samples in a pair belong to the different identity

$$FA(\tau) = \{(X_{P_1}, X_{P_2}) \in \mathcal{P}_{imp}, D_W \leq \tau\}. \quad (5)$$

Here, $TP(\tau)$ is the test samples which are classified as match pairs, whereas $FA(\tau)$ are non-match pairs which are incorrectly classified as match pairs. Both calculations are done using a single pre-defined threshold which the output distance will be compared with it as a metric for prediction.

True Positive Rate (TPR) and the False acceptance rate (FAR) are calculated as below:

$$TPR(\tau) = \frac{TP(\tau)}{\mathcal{P}_{gen}}, FAR(\tau) = \frac{FA(\tau)}{\mathcal{P}_{imp}}. \quad (6)$$

The TPR is the percentage of genuine pairs that are correctly classified as match pairs and it is called *Recall*. On the other hand, the FAR is the percentage of non-match pairs (impostor pairs) that are incorrectly classified as match pairs. According to the definitions, FAR and TPR can be computed with regard to impostor and genuine pairs, respectively.

Another metric which is called *Precision*, is widely used for accuracy evaluation. It is defined as below:

$$Precision(\tau) = \frac{TP(\tau)}{FA(\tau) + TP(\tau)}. \quad (7)$$

In comparison to Recall, Precision is the portion of retrieved positive-classified samples that are correctly classified, while recall is the portion of positive samples that are retrieved as positive. Basically, Precision and Recall are inversely correlated. The precision-recall (PR) curve demonstrates the balance between two aforementioned metrics.

Precision-Recall (PR) curves are often used in Information Retrieval and considered as an alternative to ROC curves for classification tasks with a large difference in the class distribution [37].

The main metric that has been used for performance evaluation is the Equal Error Rate (EER) which is the point when FAR and FRR are equal. Moreover, Area Under the Curve (AUC) is evaluated as the accuracy, which is the area under the ROC curve. Average-Precision (AP) is another employed metric which corresponds to the area under the Precision-Recall (PR) curve. Since we do not restrict our experiments to have the same or even close portion of genuine and impostor pairs, the AP metric can be more reliable as a representative of the accuracy which belongs to the PR curve.

For verification, the metric is simply a $\ell_2 - norm$ calculation between the outputs of the two fully connected layers from the two parallel CNNs, and a final comparison with a given threshold. To provide a better statistical demonstration of the performance, the test samples were split into 5 disjoint parts, and the averaged performance is reported across the five splits. In essence, 5-fold validation has been employed for evaluation. All performance evaluations are reported based on the statistics of the result as $\mu \pm \sigma$ using the 5-fold validation.

## VII. Training

In this section, the method and manner of training will be described. A variance scaling initializer that has been recently developed for network weight initialization [35] is used for our experiments. Batch normalization [38] has been used for improving the training convergence. The batch size is 32 for all of the experiments.

### A. Online pair selection

The pair selection in our experiments is similar to the one used in [39], in the sense of choosing challenging pairs. However, there are differences. First, and most importantly, unlike the face verification, in our experiments (AV matching), anchors cannot be defined directly as the representatives of the classes. The reason is that the pairs are defined as a one-vs-one in an audio-visual stream, and cannot be defined as a global class as can be done in image verification application. Second, since there is no anchor, the triplet loss cannot be defined as the one in [39], so we propose a method for adaptive-thresholding which will be described later in this section.

All genuine pairs are selected, and the pair selection process is restricted to only choosing main contributing impostor pairs. The criterion for choosing pairs is the distance between them in the output embedding feature space. Assume the embedding feature space is represented by $f(x) \in R^D$. If the input vector is $x$, the output is $f(x)$ with dimensionality of $D$. The no-pair-selection case is when all the impostor pairs ($X_p^{imp}$) lead to larger output distances than all the genuine pairs ($X_p^{gen}$). This leads to:

$$\left\| X_{p_i}^{imp} - X_{p_j}^{gen} \right\| > \eta, \forall (X_{p_i}^{imp}, X_{p_j}^{gen}) \in Mini - batch, \quad (8)$$

in which $\eta$ is an adaptive threshold which set a margin between impostor and genuine pairs. Algorithm 1 demonstrates the procedure.

As it can be observed from Algorithm 1, each mini-batch has its own threshold which will adaptively follow a descending order as the genuine pairs become closer on the output manifold space. Empirically we found this method to accelerate the convergence speed and moreover improved the accuracy.

### B. Hyperparameter optimization

The hyperparameters in our experiments are: $\mu$ the margin for the contrastive loss, $\lambda$ the regularization parameter, dropout parameter ($\rho$) and the $\eta_0$ as the initial margin between impostor and genuine pairs in the pair selection phase. The k-fold cross-validation method is used to estimate the hyperparameters in which we set $k = 5$.

In *k-fold* cross-validation, the original training data is randomly divided into $k$ equal parts. Of the k-parts, one of them is fixed as the validation data for testing the model, and the other $k-1$ parts are used as training data. The cross-validation

**Algorithm 1:** The adaptive online pair selection algorithm for selecting the main contributing impostor pairs in each mini-batch.

**Data**: extract mini-batch;
**Update**: Do not run optimizer (no weight update) initialization;
**Evaluation**: Feed pairs and generate output distance vector;
**Find**: Find the maximum and minimum distances belonging to genuine pairs : $max\_gen$ & $min\_gen$;
**Adaptive Threshold**: Calculate $\eta = \eta_0 \times \left|\frac{max\_gen}{min\_gen}\right|$;
**while** *checking impostor pairs* **do**
    evaluate the current impostor pair output distance: $imp\_dis$;
    **if** $imp\_dis > max\_gen + \eta$ **then**
        discard current impostor pair;
    **else**
        select current impostor pair and return its index;

process is then repeated 5 times and the average error is used to determine the best parameter.

The online pair selection is not used in the cross-validation phase. It is worth noting that the data splitting is done per subject and not just the randomly selected data. Essentially, the data is split into $k = 5$ equal parts such that none of the subjects present in one part are available in any other part. This has the practical advantage of preventing subject-specific characteristics from affecting the accuracy on the test split.

## VIII. Experiments and Results

The experiments of this section have been conducted on the audio-visual matching task to evaluate the effectiveness of the employed architecture. In evaluation of the experiments, we use the setup described in Section VII.

### A. Evaluation on LRW dataset

For audio-visual matching using the *Lip Reading in the Wild* dataset, 500 words (subjects) are available. To make the train and test sets mutually exclusive, the first 400 words are used for creating the training set and the remaining 100 words are used for test set generation. For each of the train/test sets, only 50 utterances of each word are chosen for data generation. The compiled initial training data contains generated genuine and impostor pairs. The reason this is described as initial training data is that not all the generated data is used for training. The method of selecting pairs was described in Section VII-A. The train/test characteristics are summarized in Table III.

Genuine pairs (audio/video) are created by matching the 9-channel visual feature cube with the corresponding audio feature cube as we discussed earlier in Section IV. For impostor pair generation, the audio feature map for a video is shifted alongside its time axis. The shifting is random, and could be up to 0.5-second at maximum. This shifting method allows the network to learn the matching between audio-video streams. The pair generation method is depicted in Fig. 5.

TABLE III
The train/test data summary for Lip Reading in the Wild dataset.

| set | # subjects(words) | # word utterances | # pairs |
|---|---|---|---|
| train | 400 | 50 | 280k |
| test | 100 | 50 | 70k |

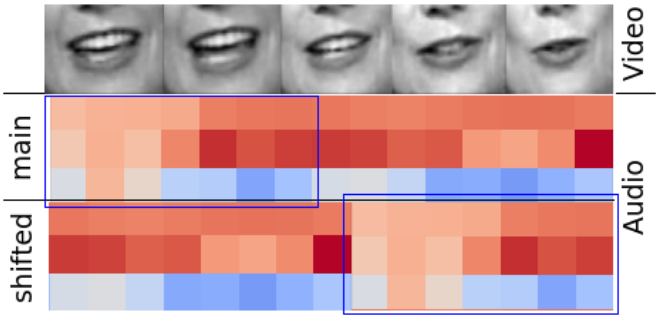


Fig. 5. Audio and video feature maps for pair generation. The shifted part is shown by the blue rectangular as an example.

Different experiments have been conducted to investigate the effects of the architecture, feature selection, and pair selection method. In all experiments performed to generate test data, unless otherwise stated, we used a 0.5-second shift for generating impostor pairs, experiments were performed using MFEC features (by using first and second order derivatives as well), and the output embedded feature space dimensionality was chosen to be 64. The training stops after 15 epochs of training data, or if the test accuracy shows descending behavior, whichever occurs first.

To demonstrate the performance of our experimental results, we report here, in tabular form, the EER and AUC values extracted from ROC curves and the AP value extracted from Precision-Recall curves.

**Effect of the proposed architecture and data representation**

In this section, the effect of choosing different data representation (MFCC/MFEC features & using temporal derivatives) on the performance results is investigated. We compare our method with the state-of-the-art in [26] in which regular CNN architecture with MFCCs for speech feature representation is used. In [26] only one channel MFCC has been used. We modify the structure represented in [26] to accept 3-channels input speech features as we use MFCC alongside with its first and second order derivatives.

For having a more comprehensive comparison, we compare our method with two common audio-visual synchrony approaches, based on CCA [28] and CoIA [40]. Prior to CCA/CoIA transformation, since the audio features are extracted at a faster rate, we interpolated the visual lip motions to have the same frame rate as audio features. Empirical evidence showed that not all the canonical correlations carry useful information. Considering the aforementioned evidence, only

20 dimensions of the correlation feature vector extracted from CCA or CoIA operations on audio-visual features are chosen which are corresponding to the higher correlation coefficients. For speech feature representation, in addition to static MFCC features, first and second order derivatives have been used as well.

The results are summarized in Table IV. The *depth* is the number of input channels, which is three if the first and second feature derivatives of MFEC/MFCC are used alongside the main features.

TABLE IV
COMPARISON OF DIFFERENT METHODS ON LRW DATASET.

| architecture | feature-depth | EER | AUC | AP |
|---|---|---|---|---|
| CCA[28] | MFCC-3 | 21.3% ± 0.88 | 85.3% ± 0.63 | 84.8% ± 0.69 |
| CoIA[40] | MFCC-3 | 20.6% ± 0.79 | 85.8% ± 0.81 | 86.0% ± 0.59 |
| CNN[26] | MFCC-1 | 17.3% ± 0.91 | 90.8% ± 0.74 | 90.2% ± 0.53 |
| CNN[26] | MFCC-3 | 17.6% ± 1.12 | 91.0% ± 1.22 | 90.4% ± 1.31 |
| 3D-CNN | MFCC-1 | 16.1% ± 0.67 | 93.7% ± 0.41 | 94.5% ± 0.72 |
| 3D-CNN | MFCC-3 | 16.3% ± 0.61 | 93.2% ± 0.46 | 94.1% ± 0.63 |
| 3D-CNN | MFEC-1 | 15.1% ± 0.52 | 94.2% ± 0.37 | 95.1% ± 0.61 |
| **3D-CNN** | **MFEC-3** | **13.5% ± 0.71** | **95.4% ± 0.36** | **96.5% ± 0.44** |

In the case of using MFCC features, using first- and second-order derivatives did not improve the performance, and additionally, using a 3-channel input increased the variance of the performance. This means that the stability of the results decreased. As can be observed in Table IV, using the MFEC feature alongside the temporal derivatives achieves the best result. We utilized our proposed *3D-CNN* architecture with different feature representations for having a better comparison solely between MFCCs and MFECs. The results demonstrated in Table IV empirically proves the effectiveness of implementing 3D architectures regardless of the feature representation since our proposed method outperforms other methods for any type of utilized features.

### Effect of embedding layer

As mentioned earlier, the default cardinality ($\zeta$) for the embedding layer is 64. In this section, we change the dimensionality of the embedding layer in order to evaluate its effect on performance, specifically, to observe the effect of feature compression. The results are shown in Table V.

TABLE V
THE EFFECT OF EMBEDDING LAYER ON PERFORMANCE.

| dim | EER | AUC | AP |
|---|---|---|---|
| 16 | 14.0% ± 0.83 | 95.0% ± 0.22 | 96.0% ± 0.51 |
| 32 | 13.8% ± 0.81 | 95.2% ± 0.42 | 96.2% ± 0.57 |
| **64** | **13.5% ± 0.71** | **95.4% ± 0.36** | **96.5% ± 0.44** |
| 128 | 13.6% ± 0.61 | 95.6% ± 0.54 | 96.3% ± 0.62 |
| 256 | 14.1% ± 0.91 | 94.9% ± 0.42 | 96.1% ± 0.48 |

The results summarized in Table V indicate that the performance changes due to variation in embedding dimensionality are not significant. However, at the highest dimensionality, a decrease can be seen. This is due to overfitting caused by using too many representative features.

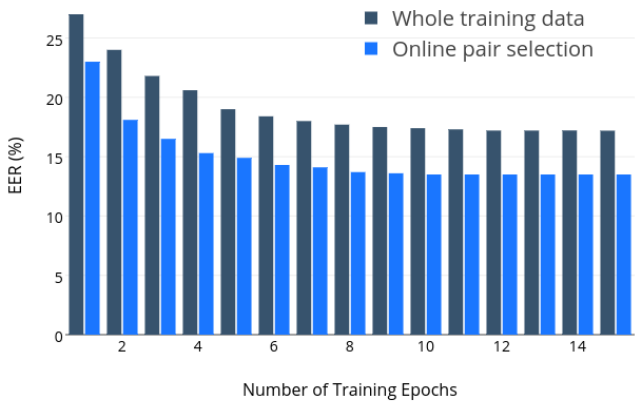


Fig. 6. The effect of the proposed adaptive online pair selection method on the speed of convergence and matching ability.

### Effect of online pair selection

The method for generating pairs has been described in Section VII-A. Here, we demonstrate the results with and without online pair selection. The setup is the default described earlier, which is using 3-channels MFEC features with the embedding dimensionality of 64. In addition to increasing the accuracy, the online pair selection resulted in a faster convergence in the training loss. Moreover, it was also faster in achieving the highest test accuracy. The EER is reported for different epochs of training for both setups. The averaged results for 5-runs of training are shown in Fig. 6, concurrently illustrating the improvement in performance and speed of convergence in which the EER belongs to evaluation on test data per number of training epochs. The effect of online pair selection on the accuracy, is illustrated in Fig. 7 for the default setup.

### The Effect of Time Shift

In this section, the effect of time shift on the performance will be investigated. However, here we do not make any changes in the training set. In this setup, we choose different shifts for generating impostor pairs solely in the test set in order to demonstrate the difficulty of the task. The results are shown in Table VI.

As it can be seen, the most challenging experimental condition is the one which has the minimum time shift. This is expected because it has increased the similarity between the genuine and impostor pairs, which has the inverse relation with the time-shifted values used for impostor pair generation.

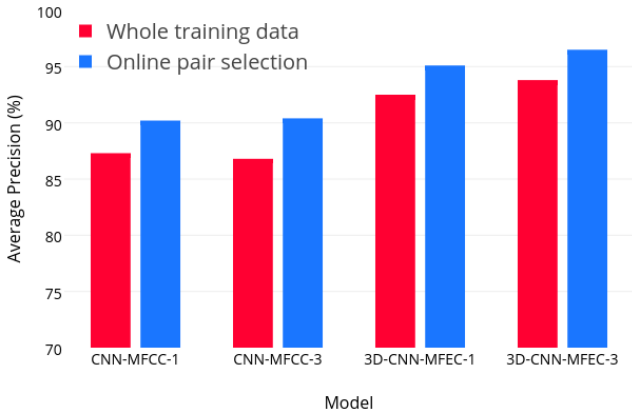


Fig. 7. The comparison of proposed adaptive online pair selection method and choosing whole training data for different architectures.

TABLE VI
THE EFFECT OF SHIFT FOR GENERATING IMPOSTOR PAIRS IN THE TEST SET

| shift(sec) | EER | AUC | AP |
|---|---|---|---|
| 0.3 | 17.3% ± 0.74 | 90.0% ± 0.68 | 89.4% ± 0.53 |
| 0.4 | 14.9% ± 0.82 | 94.0% ± 0.45 | 94.1% ± 0.61 |
| **0.5** | **13.5% ± 0.71** | **95.4% ± 0.36** | **96.5% ± 0.44** |

### B. Evaluation on the AVD dataset

The proposed method was also evaluated on the *AVD* dataset. In all experiments, the training setup is the same as described in previous experiments. For evaluation, a 0.5-second time shift is used for generating the impostor pairs. The years of 2014-2015 of the dataset are chosen for the experiments of this section. In total 495 subjects are available in years of 2014-2015. Among them, 201 subjects that are solely present in 2015 chosen to be as test subjects. The rest of the subjects are used for creating the training data.

The setup for creating pairs is within-clip data generation, e.g., the genuine and impostor pairs are built upon separated clips. Each video clip and its corresponding audio only belong to one individual. We deliberately regulate this setup for speaker-independent evaluation of audio-visual recognition.

Since the videos in the *AVD* dataset are not scripted, further data preprocessing is needed. The data preprocessing includes Voice Activity Detection (VAD) and elimination of the void sections of the visual stream in which no mouth area has been detected. The two aforementioned preprocessing operations have been performed successively, i.e., the data have been refined in two different phases. The challenge was to maintain the corresponding audio-visual streams such that they have common timing characteristics.

*Train and fine-tune on* AVD *dataset:* In this section, we first evaluate our model on the *AVD* dataset using the standard protocol of *Restricted, No Outside Data* [41]. This evaluation protocol is harsh since it assumes that no data from outside of the *AVD* dataset will be used, and the use of feature extractors that have been trained on outside data is not allowed. The results are demonstrated in Fig. 9.

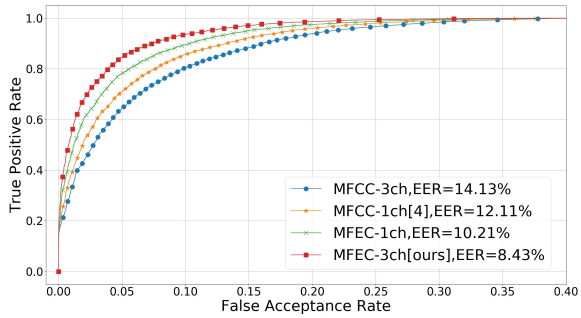


Fig. 8. The ROC curve representation for fine-tuning on *AVD* dataset.

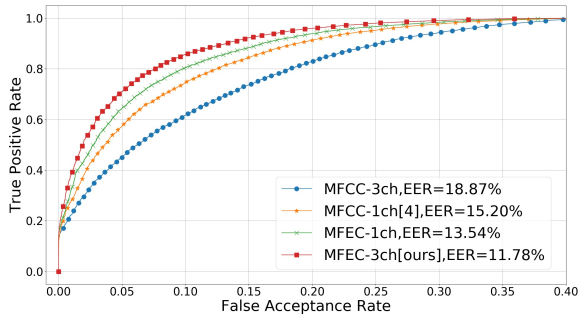


Fig. 9. The ROC curve representation for training solely on *AVD* dataset.

After experimenting using *Restricted, No Outside Data* setup, the training is done by fine-tuning the weights of the pre-trained network (Section VIII-A) by continuing the training on the *AVD* dataset. The results are depicted in Fig. 8. For the fine-tuning, the learning rate has been set to $10^{-6}$ with no decay, and the training has been performed for 15 epochs of training data. It is worth noting that using temporal derivatives for MFCC features downgraded the performance, as observed in Fig. 8. This can be related to local calculation of derivatives feature upon the non-local MFCC features. The results with comparison to other methods are summarized in Table VII.

As can be observed in Table VII, for the experiments on AVD dataset, the proposed method achieves relative improvements over 29% on the Equal Error Rate (EER) in comparison to the state-of-the-art method.

## IX. CONCLUSION

We have presented a novel coupled 3D convolutional architecture for audio-visual stream networks with convolutional fusion in temporal dimension (by utilizing 3D convolutional and pooling operations) and coupling between the networks. Experimental results on different data sets verified that the proposed architecture outperforms the other existing methods for

TABLE VII
COMPARISON OF DIFFERENT METHODS ON AVD DATASET. WE UTILIZED OUR PROPOSED ARCHITECTURE WITH DIFFERENT FEATURE REPRESENTATIONS.

| architecture | feature-depth | EER | AUC | AP |
|---|---|---|---|---|
| CCA[28] | MFCC-3 | 18.9% ± 0.83 | 90.2% ± 0.70 | 89.8% ± 0.81 |
| CoIA[40] | MFCC-3 | 18.3% ± 0.69 | 90.4% ± 0.78 | 90.1% ± 0.64 |
| CNN[26] | MFCC-1 | 12.11% ± 0.89 | 95.7% ± 0.81 | 95.9% ± 0.61 |
| CNN[26] | MFCC-3 | 14.13% ± 1.03 | 94.7% ± 1.12 | 95.0% ± 1.26 |
| 3D-CNN | MFCC-1 | 11.1% ± 0.59 | 96.2% ± 0.47 | 96.5% ± 0.70 |
| 3D-CNN | MFCC-3 | 11.3% ± 0.66 | 96.4% ± 0.51 | 96.7% ± 0.49 |
| 3D-CNN | MFEC-1 | 10.21% ± 0.62 | 96.7% ± 0.44 | 97.0% ± 0.57 |
| **3D-CNN** | **MFEC-3** | **8.43% ± 0.66** | **98.0% ± 0.41** | **98.5% ± 0.51** |

audio-visual matching, and moreover decreases the number of parameters significantly compared to the previously proposed methods. Our performance results demonstrate the effectiveness of the joint learning of spatial and temporal information using 3D convolutions rather than naively combining them within the network. The utilized local speech representative features are shown to be more promising for audio-visual recognition using convolutional neural networks.

## ACKNOWLEDGMENT

This material is based upon work supported by the Center for Identification Technology Research and the National Science Foundation under Grant #1650474.

## REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "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, Nov 2012.

[2] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 8595–8598.

[3] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *ICML*, 2011.

[4] P. Wu, H. Liu, X. Li, T. Fan, and X. Zhang, "A novel lip descriptor for audio-visual keyword spotting based on adaptive decision fusion," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 326–338, 2016.

[5] J. Huang and B. Kingsbury, "Audio-visual deep learning for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7596–7599.

[6] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[7] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, and A. Mashari, "Audio visual speech recognition," IDIAP, Tech. Rep., 2000.

[8] E. Erzin, Y. Yemez, and A. M. Tekalp, "Multimodal speaker identification using an adaptive classifier cascade based on modality reliability," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 840–852, 2005.

[9] S. Zeiler, R. Nicheli, N. Ma, G. J. Brown, and D. Kolossa, "Robust audiovisual speech recognition using noise-adaptive linear discriminant analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2797–2801.

[10] J. Wang, J. Zhang, K. Honda, J. Wei, and J. Dang, "Audio-visual speech recognition integrating 3d lip information obtained from the kinect," *Multimedia Systems*, vol. 22, no. 3, pp. 315–323, 2016.

[11] I. Almajai, S. Cox, R. Harvey, and Y. Lan, "Improved speaker independent lip reading using speaker adaptive training and deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2722–2726.

[12] S. Werda, W. Mahdi, and A. B. Hamadou, "Lip localization and viseme classification for visual speech recognition," *arXiv preprint arXiv:1301.4558*, 2013.

[13] H. Bredin and G. Chollet, "Audiovisual speech synchrony measure: application to biometrics," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 179–179, 2007.

[14] M. Slaney and M. Covell, "Facesync: A linear operator for measuring synchronization of video facial images and audio tracks," in *Advances in Neural Information Processing Systems*, 2001, pp. 814–820.

[15] G. Chetty and M. Wagner, "Liveness verification in audio-video authentication," in *Proceedings of the 10th Australian International Conference on Speech Science and Technology (SST04)*, 2004, pp. 358–363.

[16] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[17] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[18] A. Torfi, N. M. Nasrabadi, and J. Dawson, "Text-independent speaker verification using 3d convolutional neural networks," *arXiv preprint arXiv:1705.09422*, 2017.

[19] A. Torfi, "astorfi/lip-reading-deeplearning: 3D Convolutional Neural Networks for Cross Audio-Visual Matching Recognition," Aug. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.836306

[20] Y. Mroueh, E. Marcheret, and V. Goel, "Deep multimodal learning for audio-visual speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 2130–2134.

[21] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied Intelligence*, vol. 42, no. 4, pp. 722–737, Jun. 2015. [Online]. Available: http://dx.doi.org/10.1007/s10489-014-0629-7

[22] K. Noda, Y. Yamaguchi, K. Nakadai, H. Okuno, and T. Ogata, *Lip reading using convolutional neural network*. International Speech and Communication Association, 2014, pp. 1149–1153.

[23] O. Koller, H. Ney, and R. Bowden, "Deep learning of mouth shapes for sign language," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 477–483.

[24] S. Petridis and M. Pantic, "Deep complementary bottleneck features for visual speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 2304–2308.

[25] M. Wand, J. Koutnk, and J. Schmidhuber, "Lipreading with long short-term memory," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 6115–6119.

[26] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *Workshop on Multi-view Lip-reading, ACCV*, 2016.

[27] E. Marcheret, G. Potamianos, J. Vopicka, and V. Goel, "Detecting audio-visual synchrony using deep neural networks," in *INTERSPEECH*, 2015.

[28] M. E. Sargin, Y. Yemez, E. Erzin, and A. M. Tekalp, "Audiovisual synchronization and fusion using canonical correlation analysis," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1396–1403, 2007.

[29] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Asian Conference on Computer Vision*, 2016.

[30] "Audio-visual dataset, west virginia university," @ONLINE, April 2017, http://biic.wvu.edu/data-sets/.

[31] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[32] F. Developers, "ffmpeg tool (version be1d324) [software]," 2016. [Online]. Available: http://ffmpeg.org/

[33] A. Torfi, "SpeechPy: Speech recognition and feature extraction," Aug. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.810391

[34] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 1026–1034. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.123

[36] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser.

CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 539–546. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.202

[37] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 233–240. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143874

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[39] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[40] E. A. Rúa, H. Bredin, C. G. Mateo, G. Chollet, and D. G. Jiménez, "Audio-visual speech asynchrony detection using co-inertia analysis and coupled hidden markov models," *Pattern Analysis and Applications*, vol. 12, no. 3, pp. 271–284, 2009.

[41] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, pp. 14–003, 2014.

**Amirsina Torfi** received the B.Sc. in electrical engineering and the M.Sc. in information technology from Iran University of Science and Technology (IUST), Tehran, Iran, in 2011 and 2014, respectively. He is currently working toward the Ph.D. degree in electrical engineering with West Virginia University, Morgantown, WV, USA. His research interests include machine learning, image processing, speech processing and computer vision.

**Seyed Mehdi Iranmanesh** received the B.S. degree in software engineering and the M.S. degree in computer engineering (artificial intelligence) from Iran University of Science and Technology, Tehran, Iran, in 2008 and 2011 respectively. He is currently working toward the Ph.D. degree in computer science with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA. His research interests include computer vision, machine learning and vehicular networks.

**Nasser Nasrabadi** received the B.Sc. (Eng.) and Ph.D. degrees in electrical engineering from the Imperial College of Science and Technology (University of London), London, U.K., in 1980 and 1984, respectively. In 1984, he was with IBM (U.K.) as a Senior Programmer. From 1985 to 1986, he was with the Philips Research Laboratory, NY, as a member of the Technical Staff. From 1986 to 1991, he was an Assistant Professor with the Department of Electrical Engineering, Worcester Polytechnic Institute, Worcester, MA. From 1991 to 1996, he was an Associate Professor with the Department of Electrical and Computer Engineering, State University of New York at Buffalo, Buffalo, NY. From 1996 to 2015, he was a Senior Research Scientist with the U.S. Army Research Laboratory (ARL). He is currently a Professor with the Lane Department of Computer Science and Electrical Engineering at West Virginia University, Morgantown, WV. His current research interests are in image processing, computer vision, biometrics, statistical machine learning theory, sparsity, robotics, and neural networks applications to image processing. He is also a fellow of ARL and SPIE. He has served as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS, SYSTEMS AND VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON NEURAL NETWORKS.

**Jeremy Dawson, Ph.D.** has over 10 years of research experience in both academic and industrial research arenas. His current research focus is the development of devices and systems that can take advantage of the uniqueness of human molecular signatures in rapid identification scenarios. His research in the field of biosensors led to the identification of a need for new signal processing methodologies for DNA systems, which resulted in the first biomolecular biometrics (DNA) project funded through the WVU Center for Identification Technology Research (CITeR). In addition to bio-molecular biometric research, Dr. Dawson has performed multimodal biometric data collections resulting in over 12 TB of multimodal biometric data, human and bacterial DNA. He has collected data in laboratory and operational settings, and has evaluated prototype biometric sensor performance; specifically, sensor interoperability and human factors for enrollees and operators.