IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 July 2024, accepted 30 July 2024, date of publication 8 August 2024, date of current version 21 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3440646

RESEARCH ARTICLE

Arabic Lip Reading With Limited Data Using Deep Learning

ZAMEN JABR, SAULEH ETEMADI, AND NASSER MOZAYANI
School of Computer Engineering, Artificial Intelligence & Robotics Department, Iran University of Science and Technology, Tehran 13114-16846, Iran
Corresponding author: Nasser Mozayani (mozayani@iust.ac.ir)

ABSTRACT Two main challenges faced by deep learning systems are related to the amount of data and the complexity of the model concerning the number and type of layers and the number of training parameters. In this paper, we propose an end-to-end Arabic lip-reading system that can be trained on a limited dataset, which combines a visual model consisting of Convolutional Neural Networks (CNNs) and a temporal model consisting of Gated Recurrent Units (GRUs) layers, taking into account the balance between the size of the dataset and the number of model parameters. For this purpose, we created a limited Arabic dataset that involved 20 words uttered by 40 native Arabic speakers; then, we exploited the redundant frames found in video sequences to train the Arabic visemes classifier separately. This classifier was later used as a visual model, as a pre-trained model, in our end-to-end system to extract the spatial features from videos, while the temporal model was used to process the context. Our proposed method is evaluated on 1) our dataset, we obtained an accuracy equal to 83.02%; 2) the Dweik et al. dataset, we obtained an improvement rate of ≈ 3% on the result recorded by their work. In addition, we employed the visemes classifier model for person identification using the viseme shape and obtained a high result.

INDEX TERMS Arabic lip reading, deep learning, limited dataset.

I. INTRODUCTION

Lip reading can be defined as the ability to comprehend speech using only visual signal information; this process is a brilliant skill. It has many applications in speech transcription for conditions where audio signals do not exist, such as archival silent movies or off-mike conversations between politicians and personalities [1]. It is also used for persons with hearing damage, to comprehend patients with laryngeal cancer, persons who have spoken cord paralysis, and to help recognize speech in noisy environments [2].

Lip reading is a difficult process for individuals, especially when the context is unknown. Specialists need special qualities to follow their lip movements, tongue articulations, and teeth. Automatic lip-reading is also a challenging task because it requires the extraction of spatiotemporal features from a silent video, which means that both positions and motions are essential. Advances in image processing and deep learning methods have made it possible to decode this process by extracting spatiotemporal features end-to-end [2], [3].

Visual and audio-visual speech recognition methods can be classified (i) as models based on words and (ii) as those based on visemes i.e. visual units that match groups of visually indistinguishable phonemes. The earlier approach was considered more applicable to isolated word recognition, classification, and detection tasks. By contrast, the latter approach is more appropriate for sentence-level classification and continuous speech recognition with a large vocabulary [4].

According to Arabic lip-reading systems based on Deep Neural Networks (DNNs), few methods have been introduced compared with methods that use DNNs for other languages; for example, the English language. The main reason for this is the unavailability of large-scale datasets, and the acquisition of a new large-scale dataset is challenging because it is error-prone and time-consuming. A common alternative for automatic Arabic lip-reading to avoid having to train DNN from scratch is to use pre-trained models designed for other computer vision applications such as VGG-19, which was

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

used by Alsulami et al. [5], and ResNet18, which was used by Aljohani and Jaha [6].

The problem we want to solve here is how to design a DNN for Arabic Lip reading from scratch with a limited dataset size and avoid overfitting without using any pre-trained model. To this end, we consider a method to exploit the videos' repeated viseme frames to obtain sufficient data for training a deep model.

In this study, we proposed an architecture incorporating CNNs and GRUs layers, which are trained separately without requiring a large-scale Arabic dataset. The main contributions of this study are as follows.

1) Preparing a new Arabic dataset for isolated words involving the 20 words most commonly used daily.
2) Building a viseme classifier (visual model) consisting of a multilayer CNN trained on visemes extracted from words in our dataset.
3) Create an end-to-end system for Arabic word recognition where the visual model is used as the frontend and GRUs are used as the backend.
4) The same end-to-end model was applied to another Arabic dataset prepared by Dweik et al. [7] without retraining the visual model.
5) Train the same CNN architecture for person identification based on the shape of lips while speaking (visemes).

We believe that creating an Arabic lip-reading system without relying on large-scale labeled data is an important addition for artificial intelligence applications used to help persons uttering the Arabic language who have hearing impairment, where this language does not have a large-scale dataset for a lip-reading system. Another significant benefit of this system is that learning deep models on limited data opens the domain to coverage of new dialects and languages that do not have large datasets for training deep models and achieving competitive performance. It can also be used as a password verification system using lip images of spoken passwords without the need for an audio signal.

The remainder of this paper is organized as follows: Section II introduces related works, Section III explains the dataset collection process, and Section IV explains the architecture of the proposed method. The experimental results are discussed in Section V. Finally, a discussion and conclusions are presented in Sections VI and VII, respectively.

II. RELATED WORKS

There are various methods for automatic lip-reading systems. These methods can be divided into word-level and sentence-level lip-reading. In the word-level group, lip-reading is considered a classification problem, whereas in the sentence-level group, lip-reading is considered a sequence-prediction problem. The general structure of any automatic lip-reading system consists of the following steps:

1) The pre-processing step involves sampling the input video of a speaking person to image frames and extracting the Region of Interest (ROI).
2) Frontend involves extracting relevant visual features using either a handcrafted features model or a deep learning model.
3) Backend: This may be a classification model if the problem is word level or a sequence prediction model if the problem is sentence level.

In this section, we focus on recent papers on lip-reading problems for English, Arabic, and other languages based on deep learning techniques to solve this problem. We found a few lip-reading methods for the Arabic language using DNNs, mainly because of the unavailability of large-scale datasets for this language. The most important factor in the success of any DNN is the availability of a massive amount of data that is used to train a DNN. Regarding research papers that address the lip-reading problem without using DNN methods, interested people can review the references [8], [9], and [10]. We display some recent pieces of literature for automatic lip-reading methods using deep learning, where in section A, we focus on word–level methods, while in section B, we focus on sentence-level methods.

A. WORD-LEVEL METHODS

Saitoh et al. [11] produced a new method called the Concatenated Frame Image (CFI) used for sequence image representation. CFI holds spatial-temporal information of the frame sequence of an entire video. In addition, they proposed two strategies for CFI augmentation: the first strategy was applied to the spatial domain by gamma correction for brightness changes, while the second strategy was applied to the temporal domain by applying a temporal shift to control the differences in utterance speed among speakers. To implement phrase classification, they used three well-known CNNs networks (GoogLeNet [12], NIN [13], and AlexNet [14]), which were trained in other datasets unrelated to lip-reading systems and were fine-tuned for OuluVS2. The highest accuracy was 85.60% with GoogLeNet for frontal-view tests.

Chung and Zisserman [1] made two contributions: first, they built a pipeline for collecting large-scale datasets automatically from TV broadcasts, which involves more than a million-word instance uttered by over a thousand people; second, they developed CNN architectures that can recognize hundreds of word instances from their proposed large-scale dataset. They created four models based on the pre-trained VGG-M model in [15] because it performs well and is fast in classification compared to a deeper model, such as VGG-16 [16]. These Four architectures vary in how the T input frames are ingested; for intervals of one second $T = 25$. In addition, the architectures are divided based on multiple towers and early fusion, and between 2D and 3D convolutions. In the test, the set consisted of 333 words, and the top-1 accuracy was 65.4%, while the top-10 accuracy was 92.3%.

Petridis et al. [17] produced an end-to-end model for visual speech recognition, where an encoding layer consisting of three sigmoid hidden layers was joined with Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM). The encoding layer was pretrained using Restricted Boltzmann Machines (RBM). This model involves two streams: the first extracts features from the raw mouth ROI images and the second extracts features from the diff mouth ROI images. In the two streams above each encoding layer, an LSTM layer was added to model the temporal dynamics of the features. The outputs of the LSTM in each stream were concatenated and used as inputs to the Bi-LSTM layer. Thus, we can say that the model is initialized with the pre-trained encoder, while the Bi-LSTMs are trained, and the encoder parameters are fine-tuned. This model was evaluated on two datasets, OuluVS2 [18] and CUAVE [19]; the recorded accuracies were 84.5% and 78.6% respectively.

Petridis et al. [20] developed the model in [17] for application to three views of speaking persons: frontal, profile, and 45° for each view, and one stream was used to extract the features directly from the raw ROI image. The encoder layer was the same as that used in a previous study [17], where each encoder layer was followed by a Bi-LSTM to model the temporal dynamics. The output of Bi-LSTM at the three streams is concatenated in another Bi-LSTM that fuses the information output from the three streams and assigns a label for each input video frame. Their experiments were performed on the OuluVS2 dataset, where the absolute average improvement over the frontal view was 3% when two views (frontal and profile) were combined and 3.8% when three views (frontal, profile, 45°) were combined; the maximum accuracy was 96.9%.

Martines et al. [21] addressed the limitations of their model in [22], which is an end-to-end audiovisual system based on residual networks and Bi-GRUs. The processing was performed by replacing the Bi-GRU layers with Temporal Convolutional Networks (TCNs) and simplifying the training procedure by adopting a cosine scheduler [23] to execute the model training with a single stage and reduce the training time. To generalize the model to variations in sequence length, the authors proposed a variable-length augmentation method by removing random frame numbers from the video sequence and the number of removed frames from 0-5 frames. The overall model was evaluated on two datasets, LRW [1] and LRW1000 [24], for English and Mandarin, and achieved an accuracy of 85.3% and 41.4%, respectively. At the same time, the accuracy of their work [22] on the LRW dataset was 83.4%.

Mesbah et al. [2] produced a Hahn(H) CNN as a new architecture based on Hahn moments, which was used as the first layer in the CNN architecture. The reason for using discrete orthogonal Hahn moments as the first layer is to compute the moments of the input video images and hold a matrix of moments; thus, they minimize the dimensionality of the video frames and reduce the training time. In addition, to handle spatiotemporal issues in the video, they used the CFI method proposed by Saitoh et al. [11]. They evaluated their proposed model on three datasets: OuluVS2 [18], AV-Letters [25], and BBC LRW [1], with Top-1 accuracy of 59.23% for the AV-Lterrs dataset, 93.72% for the OuluVS2 dataset with five data augmentation transformation methods and speaker-independent experiments, and 58.02% for BBC LRW with one data augmentation method (flip transformation).

Fernandez-Lopez and Sukno [26] proposed a solution to train the deep model with a small data network by separating the training phase into modules, which are visual modules built using the CNN network based on the VGGM model [15], while temporal modules built using the LSTM network also introduced a method to generate weak labels per frame automatically, which are called visual units. These weak visual units guide the CNN to extract significant visual features that are combined with the context features prepared by the temporal module. The two-fold features are adequately informative for training a lip-reading system in a short time without the need for manual labeling. This system was evaluated on the OuluVS2 [18] dataset and achieved an accuracy of 91.38%.

Wang et al. [27] introduced a lip-reading method that incorporates 3D Convolution and a vision Transformer named (3DCvT). The 3DCvT is operated for spatiotemporal feature extraction of continuous images; thus, it collects local and global features in the continued images. The extracted features were sent to Bi-GRU for sequence modeling. The experiments of this method were applied to two datasets, LRW and LRW-1000 [24], and achieved 88.5% and 57.5% accuracy, respectively.

Alsulami et al. [5] introduced a lip-reading model for the Arabic language by implementing transfer learning of deep learning algorithms. They collected a new Arabic visual dataset that involved 2400 recording videos of Arabic digits and 960 video recordings of Arabic phrases uttered by 24 native Arabic speakers. The method starts with a keyframe extraction process and then uses the CFI method proposed by Saitoh et al. [11] to concatenate the keyframes of each utterance sequence in a single image. For visual feature extraction, they used a pre-trained model VGG-19 [16]. They tested different keyframe numbers: 10, 15, and 20, and compared two tactics in their proposed model: the first is the VGG-19 model alone, and the second is the VGG-19 model with the addition of a batch normalization layer. They performed their experiments on their proposed dataset, where greater accuracy was recorded with the second tactic, achieving an accuracy of 97% for phrase recognition, 94% for digit recognition, and 93% for combined digits and phrases.

Dweik et al. [7] introduced a lip-reading system for the recognition of ten Arabic words based on the image sequence equivalent of mouth movements only. They collected a dataset for these ten Arabic words from 73 native Arabic speakers who uttered those ten words once some persons uttered some of these words more than once, so the dataset obtained an overall data containing 1051 records. Three DNN models were proposed to achieve lip reading. The first model was based on CNN only, the second model was based on a combination of Time Distributed (TD)-CNN and LSTM, and the third was based on TD-CNN and Bi-LSTM. They experimented with these three models using RGB and grey image sequences. Higher results were recorded with the RGB group, where the accuracies of the CNN, TD-CNN-LSTM, and TD-CNN-Bi-LSTM were 79.2%, 70.1%, and 74.1%, respectively. In addition, the authors proposed a voting model that executes six prediction models, two cases of each DNN model, the grayscale group dataset, and the RGB group dataset. The voting model selected the highest accuracy among the six models for each word in the dataset. After applying the voting model, the overall accuracy of the lip-reading system is 82.84%.

Aljohani and Jaha [6] collected a dataset called Al-Qaida Al-Noorania Dataset (AQAND) for Arabic, comprising 10 Quranic words, 14 Quranic letters, and 29 Arabic alphabets. The AQAND was collected based on the book Al-Qaida Al-Noorania and comprised videos for 22 Arabic speakers, which were recorded from three viewpoints (0°, 30°, and 90°) for each instance in the dataset uttered three times by speakers. AQAND was used to train and test a lip-reading system based on a pretrained model (transfer learning technique). The lip-reading model in this work is based on a modified method in [28], which consists of a residual network (ResNet-18), while its backend consists of three layers of Bi-GRU followed by average pooling and fully connected layers. The authors in [28] modified the first layer in ResNet-18 from a 2D convolutional layer to a 3D convolutional layer, and the size of the kernel was 5 × 7×7. After applying pre-processing and two augmentation methods, horizontal flip and affine transformation techniques, on AQAND videos, these videos were fed into the lip-reading model. The overall accuracies of Quranic words, disjoined letters, and single letters were 83.33%, 80.47%, and 77.5%, respectively.

B. SENTENCE-LEVEL METHODS

Assael et al. [3] proposed an end-to-end model called LipNet that maps a variable-length sequence of video frames to comprehensible text. LipNet was the first model based on the end-to-end sentence level for the English language in the GRID [29] corpus. This model concurrently learns visual features using a spatiotemporal CNN (STCNN) and sequence modeling using Bi-GRUs. LipNet was trained with Connection Temporal Classification (CTC) loss [30] and recorded a recognition accuracy of 95.2% at the sentence level.

Chung et al. [31] proposed a method called Watch, Attend, and Spell (WAS) based on an encoder-decoder with an attention architecture [32], which was developed for machine translation and speech recognition. In addition, they collected a real-world dataset named LRS2, consisting of more than 100,000 sentences based on BBC television broadcasts. The performance of the WAS model on LRS2 for visual cues alone was 76.5% Word Error Rate (WER). When the WAS model was fine-tuned for other datasets, LRW [1] and GRID [29], the model yielded 23.8% and 3.0% WER, respectively.

Fenghour et al. [33] introduced a lexicon-free system that uses only visual cues (visemes) for sentence recognition. In addition, they proposed a perplexity analysis to convert recognized visemes into words. The spatial-temporal frontend was based on a model network in [4], where this network applied 3D convolution followed by a 2D Res-Net on the input image sequence. For viseme classification, they used the transformer model with an encoder-decoder structure in [34], where the encoder consisted of six self-attention layers, and the decoder consisted of three fully connected layers. Their proposed system was verified on the LRS2 dataset [31] and improved the accuracy of word classification, with a WER of 35.4%.

Sarhan et al. [35] introduced a Hybrid Lip-Reading (HLR-Net) system based on a deep convolutional neural network for lip reading from video sequences. The structure of this model was an encoder-decoder architecture, where the encoder model was built using three inception, gradient, and two Bi-GRU layers. The decoder model was built using an attention layer and a fully connected layer, and the decoder was trained using the CTC loss method [30]. On the GRID dataset [29], the HLR-Net system achieved improvements with a WER of 9.7% and CER of 4.9% in the test of unseen speakers, and a WER of 3.3% and CER of 1.4% in the test of overlapped speakers.

Peymanfard et al. [36] introduced a method that uses external text data to map visemes and characters. The authors believe that using an external model trained with textual data can improve the recognition rate for any lip-reading method. In sequence-to-sequence methods, after determining the mouth region using facial landmarks for extracting the ROIs of the image sequence, this sequence is modeled using a 3D-CNN followed by a temporal processing model where the output is a vector of probabilities for each character. However, the authors added another network model trained with independent text data. This additional model consisted of two GRU layers with an attention mechanism [37]. For the first time, an additional model was trained using textual data from the LRS2 corpus [31]. The second time, to improve accuracy, the authors used the OpenSubtitles corpus [38] to train an additional model for viseme-to-character modeling. After performing this method on the LRS2 [31] dataset, the WER improved by 4% compared to the normal sequence-to-sequence method in [31].

Fernandez-Lopez and Sukno [39] introduced a method for training an end-to-end lip-reading system using small-scale data. For this purpose, they assumed that the training of the visual front-end model should be performed in a self-supervised setting to make this model target its visemes (visual units). In addition, they presented a data augmentation method to obtain an extra temporal context by combining character-like subsequences from existing videos. The visual front-end model is based on the VGG-M [15], whereas the temporal module consists of a stack of LSTM layers. They tested their lip reading method on two scale datasets: 1) the VLRF dataset [40] for Spanish, which achieved a WER of 72.90%; 2) the TCD-TIMIT dataset [41] for the English language, which achieved a WER of 56.29%.

Kim et al. [42] introduced a speaker-adaptive lip-reading method, called user-dependent padding. The goal of this method is to treat the performance degradation problem with more lip-reading models in the case of unseen speakers. User-dependent padding is a speaker-specific input that can aid in the visual feature extraction of a pre-trained lip-reading system without adding new layers or modifying the learned weights. The authors added user-dependent padding as another input to the padding region in the CNN (frontend of the pretrained lip-reading system). User-dependent padding can be associated with convolution filters instead of traditional padding (e.g., reflect padding, zero padding, and constant padding). Thus, movement information and lip appearance are considered during visual feature encoding. In addition, to lessen the insufficiency of speaker information in the LRW dataset, the authors labeled the speakers in the LRW dataset and created a scenario for an unseen-speaker lip reading called LRW-ID based on a similar pipeline in [43]. According to the pre-trained lip-reading models, on the sentence level, the authors utilized a per-trained model of Lip-Net [3] that was applied to the GRID dataset [29] and achieved 7.2 WER; on the word level, they employed a obtained model in [21] on the LRW-ID dataset and achieved a recognition accuracy of 87.51.

El-Bialy et al. [44] introduced a system that attempts to improve lip-reading system performance using phonemes as a classification schema at the lip-reading sentence level. They investigated two classification schemas: viseme-based and character-based schemas. In this system, a spatial-temporal (3D) convolution followed by a 2D ResNet is used as a visual front-end model. A transformer with multi-headed attention was used for the phoneme recognition model. The backend model consisted of a Recurrent Neural Network (RNN), which was used as the language model. The evaluation of this system was conducted on the LRS2 dataset [31], where 70% of the phoneme recognition accuracy was achieved and 60% of the word accuracy.

As highlighted in the related works discussed in subsections A and B, the most favorable DNN architecture that has achieved the highest classification result for lip-reading systems is the combination of CNNs and RNNs (which may be LSTMs or GRUs) networks for example the works [3], [21], [26], [27], and [42]. These CNN-RNN architectures have been confirmed to be especially data-thirsty to provide good recognition accuracy because of the nature of DNN, which requires a large amount of data to train properly. In this study, we focused on constructing an end-to-end system that can perform lip-reading for the Arabic language at the word level without requiring a large-scale dataset.

Regarding previous Arabic lip-reading works, we found works [7], [5], and [6] that deal with Arabic lip reading using DNNs, where the authors in [5] and [6] depended on transfer learning from the pre-trained models for different computer vision tasks (e.g. VGG19, ResNet-18 respectively); in other words, they did not create a deep model from scratch. The authors in [7] created three-deep models from scratch and a voting algorithm but did not depend on viseme extraction. Regarding our contributions to the field of Arabic lip reading, we created from scratch two models: the first Arabic viseme classifier using deep learning, and the second is an end-to-end system for Arabic word recognition, where we used the viseme classifier model as the visual model in our end-to-end system. We also created a new Arabic dataset that involved all visemes classified by [45] and matched the 28 phonemes in the Arabic language.

Table 1 summarizes the studies on automatic lip-reading problems discussed in subsections A and B. For further reading about the lip-reading problem using DNN, refer to the surveys in [46] and [47].

III. DATASET COLLECTION

Arabic is a Hamito-Semitic language that is present in many forms. There are (1) Classical Arabic is special to the language of the Coran, (2) the Arabic dialect which differs among countries or even among regions in one country (3) Modern Classical Arabic (MCA) which is the language of education, literature, technology, science, administration and the press [45].

The problem we want to solve in this work is how to train a deep model for Arabic lip reading with the MCA form without requiring a large-scale dataset. Therefore, we propose the first step to solve this problem by creating a new small-scale dataset for isolated words. We consider the words that involve all visemes for MCA that map 28 Arabic phonemes, where this mapping was created by P. Damien [45], as shown in Table 2 displays this mapping. Our dataset consists of 20 isolated Arabic words collected from college students age range (18-25) years, all of whom were native Arabic speakers from Iraq. The laboratory environment was chosen for recording. The participant was asked to sit on a chair in front of the camera, approximately 50 cm away from the camera, where the participant's face view was frontal. Participants were asked to speak normally, as they usually speak, and leave a silence between utterances. There were 40 persons 8 Males and 32 Females. The format of the acquired data videos was mp4 recorded using a camera-type Canon with a resolution full HD (1920 × 1080 pixels) with a frame rate of 25 frames per second (fps). Each participant uttered 20 words involving Arabic digits from 1 to 10 weekdays and three other words widely used in daily life.

The selected words were uttered once by each participant, and videos of the participants were recorded in one place. The recording place was illuminated indoors, and a camera was held by a holder to reduce vibrations. The recorded videos were continuous for all datasets, which means that each participant uttered 20 words in one video during the recording process, which required additional post-processing, which is the segmentation of these recorded videos into separate sub-videos, each mapping one word in our dataset. The Free Video Cutter software is used for cutting large videos to sub-videos with one second time length per word. Figure 1 shows the participants' images in the dataset. Table 3 shows the Arabic isolated words used in preparing our dataset and their utterances and meanings in English. The dataset is available upon request.

FIGURE 1. Images of the participants in our dataset.

IV. THE PROPOSED METHOD

Our proposed method for building an Arabic lip-reading system using deep learning consists of four main stages: A) Pre-processing, B) Collecting visemes images, C) Creating an Arabic viseme classifier, and D) Creating an end-to-end model for word recognition. The four stages are explained in the following subsections. Figure 2 shows the general structure of the proposed method.

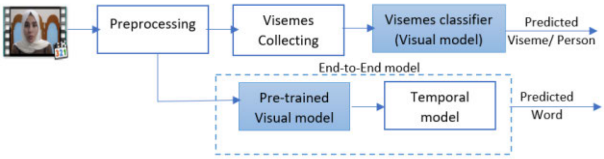


FIGURE 2. The general structure of the proposed method.

A. PRE-PROCESSING

After we collected a small-scale dataset with 20 Arabic isolated words, each video in the dataset was sampled into image frames using the OpenCV library. To extract the ROI, the frame images are processed to determine the mouth region using the facial landmark predictor from the dlip library. The facial landmark predictor attempts to determine 68 interest points that represent the face contour, eyes, nose, and mouth. Therefore, to yield the ROI that matches the talking person's lips, we used landmarks between 49 and 68 and took five rows above and below the region to capture the most uttering area, which helps to decrease the complexity and cost of computational processing when training deep models. Now, ROIs represent the visemes of each word in the dataset in the red green blue (RGB) format, as shown in Figure 3. In addition, the ROI images were resized to 112 × 112 and then rescaled (normalization of the pixel value between 0 and 1 by dividing each pixel by 255).

B. COLLECT THE VISEMES' IMAGES

To obtain a sufficient number of Arabic viseme images to train a deep CNN model (viseme classifier), we suggested exploiting the repeated visemes (similar visemes) in the ROIs of each video in our dataset by collecting all ROIs that have similar lip movement shapes to group by helping the keyframes search algorithm [5], a keyframe is a frame where a change occurs in the timeline, we took into account only five keyframes have the most variation (including the first frame because it is a reference frame in any video so we consider it the first keyframe).

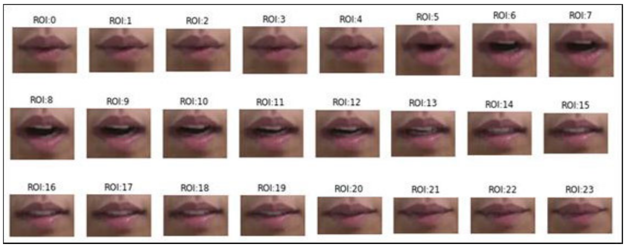


FIGURE 3. The visemes in the uttered word "واحد".

Subsequently, we manually collected similar visemes located between every two consecutive keyframes and saved them in a specific group. Each group was matched to one of the 10 classes of Arabic visemes created by Damien et al. [45]. The goal of this idea is to provide a reasonable number of visemes images to be used to train the visemes classifier. We got 9906 visemes images from 800 videos (20 words × 40 persons) in our dataset, where the silence regions in the videos were neglected. These images belong to the 10 Arabic viseme classes, which are named V1, V2,..., V10. Figure 4 shows the number of images extracted from the video sequences in our dataset. Afterward, we divided the collected viseme images into training and validation groups with a rate (of 80:20) which was collected from video words for 35 persons from our dataset, while testing was performed on unseen viseme data taken from the remaining videos for five persons in the dataset. Table 4 shows the distribution of the collected visemes in the training, validation, and testing groups.

Figures 5 and 6 show the keyframes that have the most variation on two different words ("ثلاثاء" and "واحد" respectively) and the redundant similar frames between each consecutive keyframes.

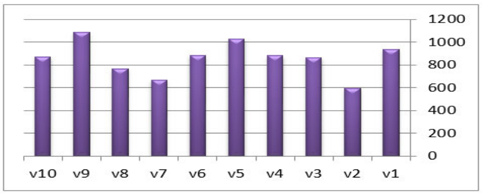


FIGURE 4. Number of extracted visemes from our dataset.

C. CREATE AN ARABIC VISEMES CLASSIFIER

When we build an Arabic visemes classifier using CNN architecture, this model will be used later as a frontend in end-to-end architecture, which we call the visual model. The

TABLE 1. Summary of some recent automatic lip-reading methods.

Reference	Year	Techniques	Dataset	Language	Re cognition task	Accuracy (Top1)	WER
T. Saitoh et al. [11]	2016	CFI+(GoogLeNet, NIN, and AlexNet)	OuLuVS2[18]	English	Word+ phrase	85.60%	-
J. S. Chung et al. [2]	2017	VGG-M	LRW[2]	English	Word	65.4%	-
Petridis et al. [17]	2017	RBM +LSTM + Bi-LSTM	OuLuVS2[18] CUAVE [19]	English	Word+ phrase Word	84.5% 78.6%	-
Petridis et al. [20]	2017	3view (RBM +Bi-LSTM) + Bi-LSTM	OuLuVS2[18] With 3 views	English	Word	96.9%	-
S. Petridis et al.[22]	2018	ResNet18 + BGRU	LRW[2]	English	word	83.4	-
B. Martines et al. [21]	2020	ResNet18 + TCN	LRW[2] LRW1000[24]	English Mandarin	Word Word	85.3% 41.4	-
A. Mesbah et al. [3]	2019	CFI+HCNN	OuLuVS2[18] AVLetters[25] LRW[2]	English English English	Word Word Word	59.23% 93.72% 58.02%	-
A.Fernandez et al. [26]	2019	VGG-M+LSTM	OuLuVS2[18]	English	Word+ phrase	91.38%	-
H. Wang et al.[27]	2022	3DCVT + Bi-GRU	LRW[2] LRW1000[24]	English Mandarin	Word Word	88.5% 57.5%	-
N. H. Alsulami et al. [6]	2022	CFI + VGG19	Custom dataset [6]	MCA Arabic	Digit+ phrase	93%	-
W. Dweik et al.[1]	2022	CNN, TD-CNN-LSTM, TD-CNN-Bi-LSTM with voting model	Custom dataset[1]	MCA Arabic	word	82.84%	-
N.F. Aljohanil et. al[7]	2023	ResNet18 +Bi-GRU	AQAND[7]	Quranic Arabic	Word Disjoint letter Single letter	83.33% 80.47% 77.5%	-
Y. M. Assael et al. [4]	2016	STCNN + Bi-GRU	GRID[29]	English	Sentence	95.2%	-
J. S. Chung et al.[31]	2020	WAS (encoder-decoder with attention)	LRS2[31] GRID [29] LRW [2]	English English English	Sentence Sentence Word	-	76.5% 3% 23.8%
S. Fenghour et al. [33]	2020	3DCNN+2D-ResNet+Transformar	LRS2[31]	English	Sentence	-	35.4%
A.M. Sarhan et al. [35]	2021	Inception + Gradient + Bi-GRU	GRID[29]	English	Sentence	-	9.7%
J. peymanfard et al. [36]	2022	GRU with attention trained on textual data	LRS2[31]	English	Sentence	-	69.5%
A.Fernandez et al. [39]	2022	VGG-M +LSTM	VLRF[39] TCD-TIMIT[41]	Spanish English	Sentence Sentence	-	72.90% 56.29%
M. Kim et al. [42]	2022	padding region in the CNN+ Lip-Net model	LRW[2] GRID [29]	English English	Word Sentence	87.51% -	- 7.2%
R. El-Bialy et al.[44]	2023	3DCNN + 2DResNet +Transformer	LRS2[31]	English	Sentence	-	40%

viseme classifier consists of four convolution layers, each followed by max-polling and batch-normalization layers, and then we add flattening and two Fully Connected layers. This model captures the spatial features of ROI images that have been resized to 112 × 112 in RGB format, the goal of using RGB format is to capture more details from the ROI image because of the difference in colors for the tongue and teeth; thus, the CNN architecture enables the extraction of mouth movement features. Table 5 presents the details of the model used for Arabic viseme classification.

D. CREATE AN END-TO-END MODEL

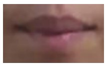
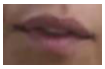
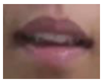
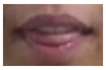
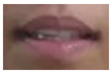
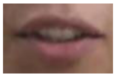
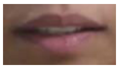
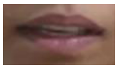
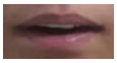
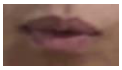
We created an end-to-end model consisting of two sub-models: a visual model to extract the spatial visual features from each video in the dataset, and a temporal model to process the temporal features of those videos and sequence modeling.

The small size of any dataset restricts the learning power of any DNN model when it has a high number of parameters [48]. Thus if we fully trained the end-to-end system with our dataset, which is a small-scale dataset, we soon realized the shortage of the dataset. Therefore, we should find a balance between the number of parameters in the end-to-end model and the amount of available training data. Therefore, we used the method for splitting the training by models proposed by [26], where our visual model involves 616,298 parameters while the temporal model involves 4,357,140 parameters, and each model is trained separately on our dataset with 800 sequences of uttering. The idea of splitting the training is beneficial to our problem, because each model has a specific aim that can be performed individually.

We proposed to employ the Arabic visemes classifier itself, which we previously trained and frozen its weights, as a visual model by removing the classification layer (Soft-max layer). Each recorded video in our dataset was passed through a pre-processing step for ROI extraction and then passed to the visual model to extract the bottleneck features (a compressed representation of the input information), which were saved as NumPy arrays.

According to the temporal model, which treats the context, we built it from a GRU layer followed by batch normalization and Bidirectional GRU layer followed by a softmax layer with 20 neurons to match several classes (words) in our dataset. The temporal model was trained separately (from scratch) on bottleneck features extracted by the visual model. The purpose of training each model separately is to balance the number of training parameters in each model with the size of the training data, reduce training time, and avoid overfitting issues. We were inspired by the idea of separating model training from Fernandez-Lopez et al. [26], but with a different architecture.

TABLE 2. Phoneme-viseme mapping for MCA language.

Viseme (V)	Phonemes in Arabic &English	Arabic viseme shape
V1	م ب [b/p] [m]	
V2	ف [f]	
V3	ض ط د ت ل ر ن [t] [d] [tˤ] [dˤ] [n] [r] [l]	
V4	ظ ذ ث [s] [sˤ] [z]	
V5	ز ص س [s] [sˤ] [z]	
V6	ج ش [ʃ] [ʒ]	
V7	ي [j]	
V8	ك غ خ [x] [ʁ] [k]	
V9	ق ه ء(ا) ع ح [ħ] [ʕ] [ʔ] [h] [q]	
V10	و [w]	

V. EXPERIMENTAL RESULTS

We discuss the results of each model separately because they were trained separately, whereas we first discuss the viseme classifier model results and then the end-to-end model results in the following subsections.

TABLE 3. Arabic words in the proposed dataset.

Arabic word	phonemes	Arabic pronunciation	Meaning in English
واحد	و ، ا ، ح، د	Wahed	One
اثنان	ا، ث، ن	Ethnan	Two
ثلاثه	ث ، ل ، ا، ه	Thalathah	Three
اربعه	ا، ر ، ب، ع ، ه	Arabah	Four
خمسه	خ ، م ، س ، ه	Khamsah	Five
سته	س ، ت ، ه	Setah	Six
سبعه	س ، ب ، ع ، ه	Sabaah	Seven
ثمانيه	ث ، م، ا، ن ، ي ، ه	Thamanyah	Eight
تسعه	ت، س ، ع ، ه	Tesaah	Nine
عشره	ع ، ش ، ر ، ه	Ashraah	Ten
سبت	س ، ب ، ت	Sabt	Saturday
أحد	ا ، ح ، د	Ahad	Sunday
إثنين	ا ، ث ، ن ، ي	Ethnain	Monday
ثلاثاء	ث ، ل، ا ، ء	Thulathaa	Tuesday
أربعاء	ا، ر، ب، ع،ء	Arbeaa	Wednesday
خميس	خ ، م ، ي ، س	Khamees	Thursday
جمعه	ج، م ، ع ، ه	Jumaa	Friday
متى	م ،ت ، ا	Mataa	When
كيف	ك، ي ، ف	Kayfa	How
اسف	ا، س ، ف	Asef	Sorry

TABLE 4. Distribution of the collected visemes.

Training samples	Validation samples	Testing samples
6853	1720	1333

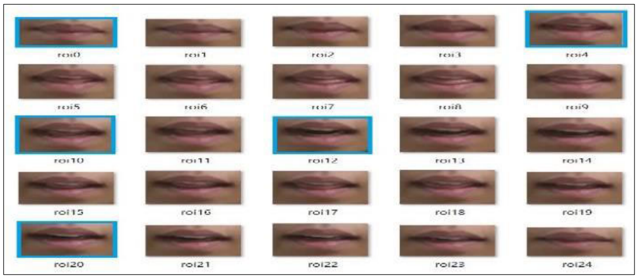


FIGURE 5. Highlighted five keyframes have the most variation in the word "ثلاثاء".

A. VISEMES CLASSIFIER RESULTS

The proposed visemes classifier model was trained using the Adam optimizer for 100 epochs with a learning rate starting with 0.0003 and decreasing factor Sqrt (0.1) with an average patience of 5 epochs and a batch size equal to 16. The image-data-generation class was used to artificially augment the size of our training data, thereby supplying many more images to the model for training. We performed image augmentation forms involving horizontal flip, rotation with 5 degree,

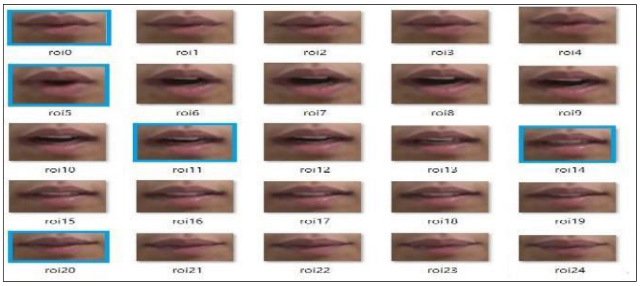


FIGURE 6. Highlighted five keyframes that have the most variation in the word"واحد".

TABLE 5. Details of the visemes classifier.

Layer type	Output shape	Parameters
Input Layer	(None, 112, 112, 3)	0
Conv2D	(None, 108,108,256)	19456
Activation	(None, 108,108,256)	0
Maxpooling2D	(None,54,54,256)	0
Batch Normalization	(None,54,54,256)	1024
Conv2D	(None,52,52,128)	295040
Activation	(None,52,52,128)	0
Maxpooling2D	(None,26,26,128)	0
Batch Normalization	(None,26,26,128)	512
Conv2D	(None,24,24,64)	73792
Activation	(None,24,24,64)	0
Maxpooling2D	(None,12,12,64)	0
Batch Normalization	(None,12,12,64)	256
Conv2D	(None, 10, 10, 32)	18464
Activation	(None, 10, 10, 32)	0
Maxpooling2D	(None, 5, 5, 32)	0
Batch Normalization	(None, 5, 5, 32)	128
Flatten	(None,800)an	0
Dense	(None,256)	205056
Dense	(None,10)	2570

horizontal and vertical shifting, change the brightness in range (1-2), and shear the image in the range 0.2. The visemes classifier model yielded classification accuracies of 91.14, 86.22, and 86.87% in the training, validation, and testing groups, respectively. Figure 7 shows the confusion matrix for the test group.

In addition, we made the same proposed CNNs model for the Arabic visemes classifier to have the ability to perform person identification task by uttering visemes with good results. This task is performed with some simple changes: learning rate = 0.0001 and decreases factor equal to Sqrt (0.1) with an average patience of 5 epochs, number of epochs = 50, patch size = 16, and number of classes in the output layer = 40, which match the number of persons in our dataset. CNNs model training was performed using the Adam optimizer. The numbers of visemes images used in the training, validation, and testing groups were 6658, 816, and 870, respectively. We obtained an accuracy of 99.77% in person identification, as shown in the confusion matrix in Figure 8. Thus, we conclude that the viseme image can be used as a biometric measure for person identification, where the selection of viseme images that are used in training and testing does not belong to the same letter in Arabic but from different letters (arbitrary selection for viseme images). To the best of our knowledge, this Arabic viseme classifier is the first in-field classification for Arabic visemes using a DNN. Table 6 shows the classification report for the testing group. The accuracy and loss curves during the training phase are shown in Figure 9.

As shown in Figure 9, the training vs. validation accuracy curves for the visemes classifier model began to improve at epoch 45 and reached a peak at epoch 100, where model training was stopped, and the training accuracy was close to 91.14% for the training group and 86.22% for the validation group. In contrast, the training and validation loss curves decreased to a point of stability and did not have a large gap between them at epoch 45 and were close to zero at epoch 100, where the model training and validation processes were stopped to avoid overfitting. This means that the difference between the predicted visemes and actual visemes is low, and as a result, we obtained better model performance.

If we look closely at the confusion matrix in Figure 7 and the classification report in Table 6, we notice that the high recognition accuracy recorded for V1 = 0.97, V4 = 0.94, V5 = 0.90, V6 = 0.97, and V10 = 0.95, the reason for the high accuracy of these visemes is the distinct shapes that do not overlap with other shapes, so that the model can recognize them well. Where V1 results from the pursed lips, V4 results from the appearance of the tongue outside the mouth, V5 results in the proximity of the teeth of the upper and lower jaws and the flattening of the upper and lower lips, V6 results from the teeth of the upper jaw appearing aligned between the upper and lower lips with the lips protruding upward, and V10 results from pursed lips with a dark hole between the lips (see phoneme-viseme mapping in Table 2). V7 and V8 have lower accuracy (0.66 and 0.51, respectively) because their shapes overlap by uttering the words, and we can see that clearly in the confusion matrix, 12 of 77 images of V7 were incorrectly classified as V8, and there were 10 of 53 images of V8 incorrectly classified as V7. Moderate accuracy was recorded for V2, V3, and V9, with rates of 0.67, 074, and 0.88, respectively.

From the confusion matrix of the person identification model in Figure 8, we do not notice any overlap among person recognition predictions because each person has a distinct lip shape, which can be used as a biometric measure for person identification, where we obtained an accuracy rate of 99.77%. At the same time, this distinct shape makes the problem of lip reading more difficult because each person utters the same word with different lips shapes and speeds.

B. END-TO-END MODEL RESULTS

After we completed training the Arabic viseme classifier (visual model) and obtained an acceptable result in the viseme classification task, we froze its weights and saved this model. The last layer in this model (Soft-max layer) was removed, and this model was used as a pre-trained model that can be used as a frontend in our end-to-end model. The visual model

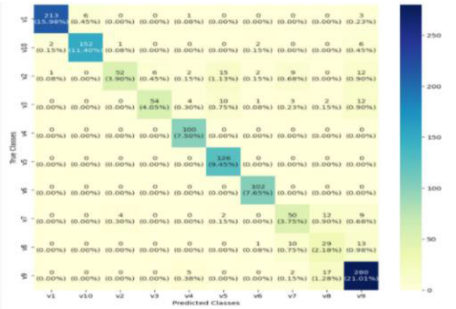


FIGURE 7. Confusion matrix for viseme classifier model.

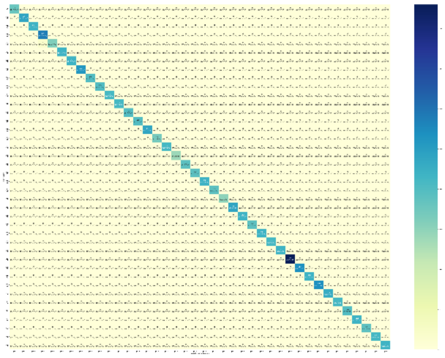


FIGURE 8. Confusion matrix for person identification model.

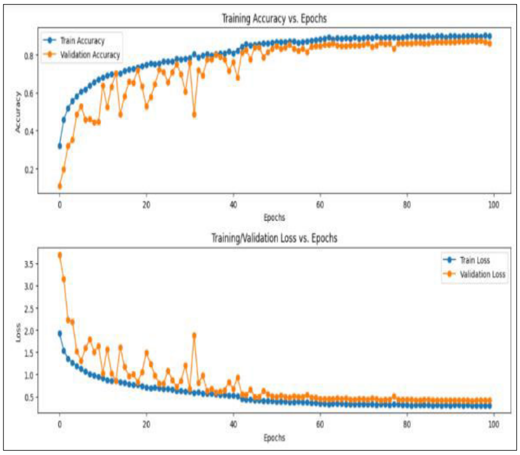


FIGURE 9. The accuracy and loss curves during the training of the visemes classifier model.

extracts bottleneck features for each video in the dataset and saves them as NumPy arrays.

To obtain more training data for the temporal model, we applied augmentation mechanisms: rotation, sigmoid, flip, and linear transformation for each video in the dataset, and then extracted their bottleneck features using the visual model. The total number of videos used in training equals the number of persons in the training group (32) × number of words uttered (20) = 640 videos. After applying augmentation (rotation, sigmoid, flip, and linear), we got 3200 videos.

TABLE 6. Classification report for arabic visemes classifier model.

Classes	Precision	Recall	F1-score	Support
V1	0.99	0.96	0.97	223
V2	0.91	0.53	0.67	99
V3	0.90	0.63	0.74	86
V4	0.89	1.00	0.94	100
V5	0.82	1.00	0.90	126
V6	0.94	1.00	0.97	102
V7	0.68	0.65	0.66	77
V8	0.48	0.55	0.51	53
V9	0.84	0.92	0.88	304
V10	0.96	0.93	0.95	163
Accuracy			0.87	1333
Macro_avg	0.84	0.82	0.82	1333
Weighted_avg	0.87	0.87	0.86	1333

Some persons uttered the same word more than once, which means that we obtained extra data, so the overall summation of the training video is that 3289 videos will be entered into visual models. The validation group comprised 20% of the training group, with 658 videos; thus, the remaining number for training was 2631. The number of videos in the testing group (unseen data) was 8 persons ×20 words uttered =160 videos (one video was neglected because it was damaged, so the total number of videos in the testing group was 159), all of which were entered into the visual model to extract the bottleneck features. The temporal model consisted of one GRU layer with 512 neurons with a time step of 25, followed by a batch normalization layer and one Bi-GRU layer with 1024 neurons. This model was trained on bottleneck features using the Adam optimizer for 70 epochs with a batch size of 16 and a learning rate of 0.0003 to classify feature arrays into one of 20 classes at the softmax layer. The recorded accuracy result was 83.02.

Also, we applied our proposed models to another Arabic dataset created by W. Dweik et al. [7] which involved 10 Arabic words (‘‘اسف,’’ ‘‘اليوم’’,‘‘غدا’’, ‘‘جميل’’, ‘‘خير’’, ‘‘مرحباً’’, ‘‘مساءً’’, ‘‘صباحاً’’, ‘‘سلام’’, ’’شكراً‘‘) these words are used frequently in the day and are uttered by 73 native Arabic speakers in mp4 format with rat 30 fps, we also applied the same augmentation methods that applied on our dataset. Here, we did not need to retrain the visual model again; instead, we used it as a pertained model on Arabic visemes that can extract visual features directly from a new dataset. Subsequently, in the temporal model, we only changed the time step to 30 and the softmax layer to 10 classes to match the number of classes in this dataset, after which we trained the temporal model on bottleneck features using the Adam optimizer for 70 epochs with a batch size of 16. We obtained a good accuracy of 85.81 as shown in Table 7, where the recorded results indicate that our proposed method gave recognition results that exceeded the results recorded by Dweik et al. [7] with an improvement rate equal to approximately 3%. Figures 10 and 11 show the train vs. validation accuracy and loss curves by the proposed end-to-end model for our dataset and the W. Dweik dataset, respectively, whereas Figures 12 and 13 show confusion matrices for testing groups on the two datasets. -

As shown in Figures 10 and 11, the training vs. validation accuracy curves for the end-to-end model began to improve at epoch 40 and peaked at epoch 70, where model training was stopped and the training accuracy was close to 100 in our dataset experiment. On the Dweik dataset, the model accuracy improved at epoch 20 and peaked at epoch 70, where model training was stopped and the training accuracy was close to 100. We believe that the improvement in accuracy began from epoch 20 because the number of classes in the W. Dweik dataset was less than the classes on our dataset with a rate of half. The training and validation loss curves decreased to a point of stability and did not have a large gap between them at epoch 20 on our dataset and epoch 10 in the Dweik dataset, and close to zero at epoch 70 in both datasets, where the model training and validation processes were stopped to avoid overfitting. This means that the difference between predicted and actual words is low.

If we look closely at the confusion matrixes in Figure 12 and the classification report in Table 8 we will find that the highest recognition rate of 100% recorded on words (“Khamsah, Setah”, “Asef”, “Ashraah”, “Mataa”), another high recognition rate is 80-90% recorded in words (“Thamanyah,” “Arabah,” “Jumaa,” “Kayfa,” “Tesaah,” “Wahed,” “Sabaah,” “Ahad,” “Khamees,” “Arbeaa”), moderate rate 60-70% recorded on the words (“Sabt,” “Thulathaa”), low recognition rate< 60% recorded on words (“Thalathah,” “Ethnan,” “Ethnain”) because the word “Thalathah” overlap with the words “Thulathaa” with rate 25% in confusion matrix because those words very simalar in uttering (they have more similar phonemes see Table3), the word “Ethnan” overlap with the words “Ethnain” with a rate of 12.5% because those words very simalar in uttering (they have very similar phonemes) and with rate of 12.5% for words (“Thamanyah,” “Thalathah”) because those words begin with phonems “Tha” (“ث”) wich is the more stress in those three words (“Thamanyah,” “Thalathah,” “Ethnan”), so the model confused on recognition among those three words. The word “Ethnain “overlaps with the words “Ethnan” with a rate of 25% in the confusion matrix because those words are very similar in uttering (they have more similar phonemes) and with a rate of 12.5% with the word “Thalathah “ because those words have the phonemes “Tha” (“ث”) which is the more stress in those two words (“Thalathah,” “Ethnain”).

If we look closely at the confusion matrixes in Figure 13 and the classification report in Table 9 we will find that the highest recognition rate recorded on words (“Masaa”, “Alyawm”, “Salam”, and “Shukrun”) with rates (91%, 92%, 93%, and 93%) respectively. The moderate accuracy range was recorded on the words (“Jameel”, “Sabah”, Aasef, and “Marhaba”,) (84%, 85%, 86%, and 88%) respectively. The lowest result was recorded on the words (“Ghadan,” and “Khair”) with rates (of 74%, and 75%) because these two words begin with the same visemes but different phonemes (“غ,” “خ” = V8 see Table 2) and the stress done on these two phonemes through uttering them so the confusion got in the recognition where the word “Khair” overlapped with the word “Ghadan” with a rate 20.6%, while the word “Ghadan” overlapped with the word “Khair” with a rate 10% as it is clear from confusion matrix in Figure 13. Note that we find the overlap rates by dividing the number of misclassified words in a specific class by the total number of samples from this word.

If we notice the words in the Dweik dataset we don’t find two words that have more identical phonemes on the contrary, our dataset has 4 pairs of words that have more identical phonemes (as a result they have similar visemes) which makes the recognition task difficult, these pairs (“Ethnan”, “Ethnain”), (“Thalathah“, “Thulathaa”,), (“Arabah”, “Arbeaa”), and (“Khamsah”, “Khamees”). Also, the W. Dweik dataset didn’t involve all visemes in the Arabic language, for example, V4 matches three phonemes “ذ”, ”ظ”, and “ث” unfound in this dataset while our dataset comprises all visemes classified by P. Damien [45].

Let us compare the results of our proposed method with those of the traditional method proposed by Damien [49] based on feature engineering for Arabic viseme images. We found that our method has the best result, with approximately 2% for word recognition and 3% for viseme recognition. The method in [49] is the only work based on visemes for Arabic word recognition before our work, which uses the geometrical features of predefined visemes and the Hidden Markov Model (HMM) as a classifier. Training and testing were performed on a custom dataset with an MCA form prepared by the author involving 20 words uttered by four persons that were recorded in a controlled environment. The accuracy of viseme recognition was 83.92% for 852 Arabic visemes with 10 classes, as classified by Damien et al. [45], while the accuracy of word recognition was 81.67% for overall 240 words, where the author did not indicate the number of words used in training and testing.

Based on these experimental results, we proved that the visual model can be used as a pre-trained model to extract features from different datasets and provide good accuracy without the need to train it from scratch. To the best of our knowledge, this is the first study to use the DNNs in Arabic visemes classification. The test experiments were performed by independent persons (the visemes and words uttered by unseen persons on training data) in both the visemes classifier and the end-to-end models. We executed these models on a computer with an Intel(R) Core(TM) i7-10750H processor (2.60 GHz), 32 GB RAM, and a single NVIDIA GeForce RTX 3060 graphic processing unit. The proposed models were implemented using the Keras framework with TensorFlow backend.

VI. DISCUSSION

For automatic lip reading at the word level using DNNs, researchers have presented many methods that mostly rely

TABLE 7. The accuracy results of the proposed end-to-end model.

Method	Dataset	Number of classes	Number of Speakers	Train/val./test Groups	Techniques	Accuracy
W. Dweik et al. [1] in 2022	Custom dataset with MCA form prepared by the authors [1]	10	73	1304/250/274 (with augmentation on training group)	Three DNNs with a voting model	Without voting =79.2% With voting= 82.84%
Our proposed method	**Our dataset with MCA form**	20	40	2631/658/160 (with augmentation on training group)	End-to-end model based on visemes classifier	**83.02%**
Our proposed method	W. Dweik dataset [1]	10	73	3196/564/275 (with augmentation on training group)	End-to-end model based on visemes classifier	**85.82%** Improvement≈**3%**

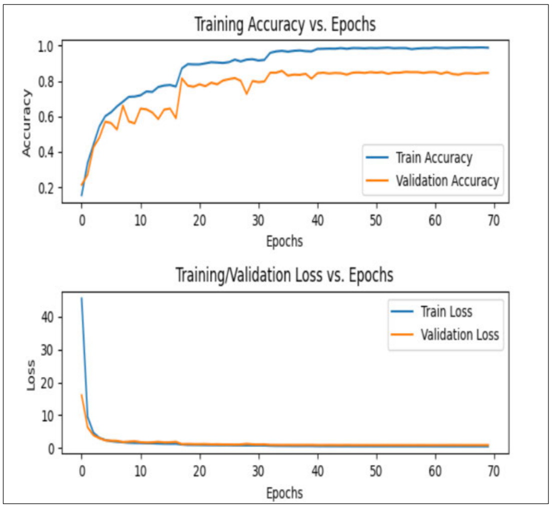


FIGURE 10. The accuracy and loss curves of the end-to-end model on our dataset.

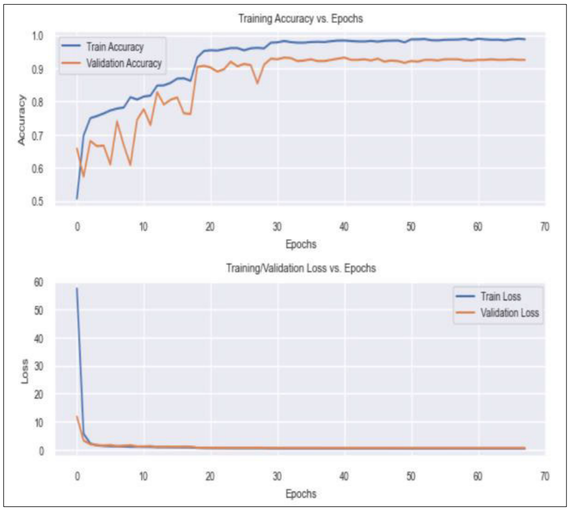


FIGURE 11. The accuracy and loss curves of the end-to-end model on the Dweik dataset.

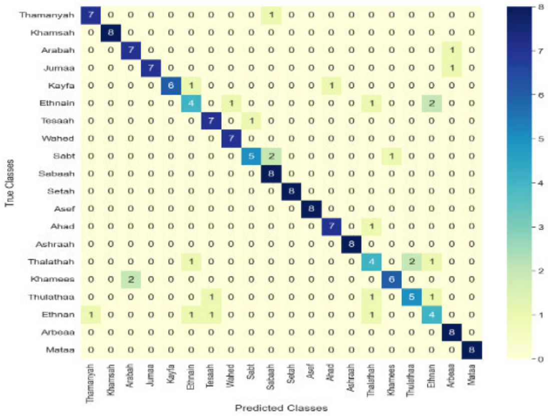


FIGURE 12. Confusion matrixes of the testing group on our dataset.

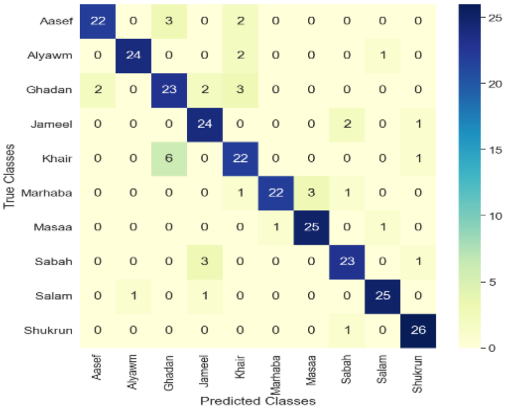


FIGURE 13. Confusion matrixes of the testing group on the W. Dweik dataset.

on large-scale datasets, such as those in [1], [21], [22], and [27]. However, if we want to retrain these methods for any limited datasets, we notice a large decrease in recognition accuracy rates because of a shortage of data and an imbalance between the number of parameters of the model and the size of available training data, which we noticed when we tried to apply a DNN architecture similar to the work in [1] on our proposed dataset. Therefore, we must find an alternative strategy for training the DNN on our limited dataset. However, the limited size of our dataset and the large number of DNN model parameters are two challenges; how we solve and find a balance between them. To this end, we attempted to find a way to construct a new visual model specialized in the problem of Arabic lip-reading with fewer parameters to fit our limited dataset. This new model can be used as a frontend in our end-to-end model, which is constructed for word recognition, where the two models are trained separately.

Dataset availability is an important limitation of Arabic lipreading systems. Moreover, acquiring a new large-scale dataset is challenging, particularly because of the need for suitable labeling, which is time-consuming and error-prone. A common alternative to avoid training the DNN from scratch is to use pre-trained models designed for other computer vision applications, such as VGG-19, which was used by Alsulami et al. [5], and ResNet18, which was used by Aljohani and. Jaha [6]. Thus, we propose the creation of an end-to-end system from scratch trained on a limited dataset for the Arabic language with MCA form, then exploiting the repeated visemes available in recorded video sequences. These visemes are used for training the deep CNN model (viseme classifier), which will be later used as frontend in our end-to-end model.

To avoid the overfitting problem, we took several countermeasures. First, we exploited the repeated visemes in each word in the dataset to collect a reasonable number of visemes images (see Figure 4), which were used to train the visemes classifier model. Second, we used four types of augmentation (flip, rotate, linear, and sigmoid) on each video in the training and validation groups of the dataset. Third, L2 weight regularization (0.05 is used in the visual model and 0.03 in the temporal model) to encourage parameter sparsity and penalize negative or highly positive weights. The fourth batch normalization was performed after each convolutional layer in the visual model, and only after the GRU layer in the temporal model.

Because the problem in this study is a classification, three metrics Precision, Recall, and F1-score were adopted to evaluate the model's performance of prediction for each class in the visemes dataset and each word in two datasets (our and the Dweik et al. datasets) as shown in Tables 6, 8, and 9 where Precision calculates how often the proposed model correctly predicts positive cases, Recall represents how well the model can identify actual positive cases, and F1-score is the weighted mean of Precision and Recall. The three metrics are shown in equations 1-3, where TP, FP, and FN represent True Positive, False Positive, and False Negative respectively. If we show values of the F1-score in three Tables 6,8, and 9 we notice most of these values are high, which means a well-balanced performance for both the visemes classifier and the end-to-end models. In Table 6 the higher F1-score recorded equals 0.97 for both visemes V1 and V6. In Table 8 the higher F1-score recorded equals 1.00 for the words ‘‘Khamsah’’, ‘‘Setah’’, ‘‘Asef’’, ‘‘Ashraah’’, and ”Mata”. In Table 9 the higher F1-score recorded equals 0.93 for both words ‘‘Salam’’ and ‘‘Shukrun’’.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1} - \text{score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{3}$$

VII. CONCLUSION AND FUTURE WORK

Our proposed end-to-end system consists of two models: a visual model (viseme classifier) and a temporal model, which are trained separately because each model has a specific aim that can be reached individually. The visual model aims to determine the visual information that is observable at a given instant. Simultaneously, the temporal module maps the visual features into speech units while joining temporal constraints. Based on the experimental results, we conclude the following:

a. The better the performance of the visual model, the better the results in the temporal model.

b. The proposed visual model can be used to extract bottleneck features from any Arabic dataset independent of the speaker or type of word without needing to train the visual model from scratch. We only need to retrain the temporal model and change its softmax layer to fit the new classes of the required dataset. We proved that when we applied our proposed method to the dataset created by Dweik et al. [7], we noticed an increase in the recognition rate by approximately 3% (see classification report in Table 9). We did not need to retrain the visual model from scratch, we used it as a pre-trained model to extract the visual features for each video. These features train the temporal model to learn to recognize an entire word. This procedure reduces training time and simplifies the work on the overall system.

c. We can avoid the overfitting issue that arises in lip-reading systems using DNNs when the size of the dataset is limited by exploiting the repeated viseme frames and using them to train a visual model.

d. The proposed end-to-end model succeeded in recognizing very similar articulation words in our dataset with an acceptable result such as the words ((‘‘Ethnan,’’ ‘‘Ethnain’’), (‘‘Thalathah,’’ ‘‘Thulathaa,’’), (‘‘Arabah,’’ ‘‘Arbeaa’’), and (‘‘Khamsah,’’ ‘‘Khamees’’) as seen in Table 8 that displays the classification report for this model on our prepared dataset.

e. Regarding computational complexity (time, space, and number of training parameters), we noticed a reduction in model training time, where the total training time for the overall end-to-end model was approximately 3.57 hours (≈ 2.7 hours for the visual module), without requiring any pre-trained model (model designed for other computer vision applications) or external training data. According to space, we needed to execute our proposed models on 32 GB RAM and a single NVIDIA GeForce RTX 3060 graphic processing unit to speed

TABLE 8. Classification report on our dataset.

Word	Precision	Recall	F1-score	Support
Thamanyah	0.88	0.88	0.88	8
Khamsah	1.00	1.00	1.00	8
Arabah	0.78	0.88	0.82	8
Jumaa	1.00	0.88	0.93	8
Kayfa	1.00	0.75	0.86	8
Ethnain	0.57	0.50	0.53	8
Tesaah	0.78	0.88	0.82	8
Wahed	0.88	1.00	0.93	7
Sabt	0.83	0.62	0.71	8
Sabaah	0.73	1.00	0.84	8
Setah	1.00	1.00	1.00	8
Asef	1.00	1.00	1.00	8
Ahad	0.88	0.88	0.88	8
Ashraah	1.00	1.00	1.00	8
Thalathah	0.50	0.50	0.50	8
Khamees	0.86	0.75	0.80	8
Thulathaa	0.71	0.62	0.67	8
Ethnan	0.50	0.50	0.50	8
Arbeaa	0.80	1.00	0.89	8
Mataa	1.00	1.00	1.00	8
accuracy			0.83	159
macro avg	0.83	0.83	0.83	159
weighted avg	0.83	0.83	0.83	159

up the execution. If we noticed the training parameters, the visual model had 616,298 parameters while the temporal model involved 4,357,140 parameters, where the number of parameters in the visual and temporal model was much less than pre-trained models such as VGG16 or ResNET18; therefore, we obtained a type of balance between the size of the dataset and the number of training parameters and enabled us to avoid overfitting problems.

f. We succeeded in using the visual model to perform another task, person identification based on viseme shape, and achieved high accuracy.

TABLE 9. Classification report on the W. Dweik dataset.

Word	Precision	Recall	F1-score	Support
Aasef	0.92	0.81	0.86	27
Alyawm	0.96	0.89	0.92	27
Ghadan	0.72	0.77	0.74	30
Jameel	0.80	0.89	0.84	27
Khair	0.73	0.76	0.75	29
Marhaba	0.96	0.81	0.88	27
Masaa	0.89	0.93	0.91	27
Sabah	0.85	0.85	0.85	27
Salam	0.93	0.93	0.93	27
Shukrun	0.90	0.96	0.93	27
accuracy			0.86	275
macro avg	0.87	0.86	0.86	275
weighted avg	0.86	0.86	0.86	275

g. In future work, we will first collect a new dataset for sentences in Arabic. Second, we attempted to develop the proposed method for sentence-level prediction.

ACKNOWLEDGMENT

The authors would like to extend their sincere thanks and appreciation to all the participants who volunteered in the Arabic dataset acquisition phase.

REFERENCES

[1] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Proc. Asian Conf Comput. Vis. (ACCV)*, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Cham, Switzerland: Springer, 2017, pp. 87–103, doi: 10.1007/978-3-319-54184-6_6.

[2] A. Mesbah, A. Berrahou, H. Hammouchi, H. Berbia, H. Qjidaa, and M. Daoudi, "Lip reading with Hahn convolutional neural networks," *Image Vis. Comput.*, vol. 88, pp. 76–83, Aug. 2019, doi: 10.1016/j.imavis.2019.04.010.

[3] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, "LipNet: End-to-end sentence-level lipreading," 2016, *arXiv:1611.01599*.

[4] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with LSTMs for lipreading," in *Proc. Interspeech*, Aug. 2017, pp. 3652–3656, doi: 10.21437/interspeech.2017-85.

[5] I. Ullah, H. Zahid, F. Algarni, and M. A. Khan, "Deep learning-based approach for Arabic visual speech recognition," *Comput., Mater. Continua*, vol. 71, no. 1, pp. 85–108, 2022, doi: 10.32604/cmc.2022.019450.

[6] N. Faisal Aljohani and E. Sami Jaha, "Visual lip-reading for quranic Arabic alphabets and words using deep learning," *Comput. Syst. Sci. Eng.*, vol. 46, no. 3, pp. 3037–3058, 2023, doi: 10.32604/csse.2023.037113.

[7] W. Dweik, S. Altorman, and S. Ashour, "Read my lips: Artificial intelligence word-level Arabic lipreading system," *Egyptian Informat. J.*, vol. 23, no. 4, pp. 1–12, Dec. 2022, doi: 10.1016/j.eij.2022.06.001.

[8] W. U. R. Butt and L. Lombardi, "A survey of automatic lip reading approaches," in *Proc. 8th Int. Conf. Digit. Inf. Manage. (ICDIM)*, Sep. 2013, pp. 299–302, doi: 10.1109/ICDIM.2013.6694023.

[9] Z. Zhou, G. Zhao, X. Hong, and M. Pietikäinen, "A review of recent advances in visual speech decoding," *Image Vis. Comput.*, vol. 32, no. 9, pp. 590–605, Sep. 2014, doi: 10.1016/j.imavis.2014.06.004.

[10] S. Mathulaprangsan, C.-Y. Wang, A. Z. Kusum, T.-C. Tai, and J.-C. Wang, "A survey of visual lip reading and lip-password verification," in *Proc. Int. Conf. Orange Technol. (ICOT)*, Dec. 2015, pp. 22–25, doi: 10.1109/ICOT.2015.7498485.

[11] T. Saitoh, Z. Zhou, G. Zhao, and M. Pietikäinen, "Concatenated frame image based CNN for visual speech recognition," in *Proc. Asian Conf. Comput. Vis.*, Taipei, Taiwan, 2016, pp. 277–289, doi: 10.1007/978-3-319-54427-4_21.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[13] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, 2014, doi: 10.48550/arXiv.1312.4400. [Online]. Available: https://dblp.org/db/conf/iclr/iclr2014.html

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[15] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2014. [Online]. Available: https://arxiv.org/abs/1405.3531

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

[17] S. Petridis, Z. Li, and M. Pantic, "End-to-end visual speech recognition with LSTMS," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2592–2596, doi: 10.1109/ICASSP.2017.7952625.

[18] I. Anina, Z. Zhou, G. Zhao, and M. Pietikäinen, "OuluVS2: A multi-view audiovisual database for non-rigid mouth motion analysis," in *Proc. 11th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit. (FG)*, vol. 1, May 2015, pp. 1–5, doi: 10.1109/FG.2015.7163155.

[19] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy, "CUAVE: A new audio-visual database for multimodal human–computer interface research," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, vol. 2, May 2002, pp. 2017–2020, doi: 10.1109/ICASSP.2002.5745028.

[20] S. Petridis, Y. Wang, Z. Li, and M. Pantic, "End-to-end multi-view lipreading," 2017, *arXiv:1709.00443*.

[21] B. Martinez, P. Ma, S. Petridis, and M. Pantic, "Lipreading using temporal convolutional networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6319–6323, doi: 10.1109/ICASSP40776.2020.9053841.

[22] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic, "End-to-end audiovisual speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 6548–6552, doi: 10.1109/ICASSP.2018.8461326.

[23] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2017, *arXiv:1608.03983*.

[24] S. Yang, Y. Zhang, D. Feng, M. Yang, C. Wang, J. Xiao, K. Long, S. Shan, and X. Chen, "LRW-1000: A naturally-distributed large-scale benchmark for lip reading in the wild," in *Proc. 14th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2019, pp. 1–8, doi: 10.1109/FG.2019.8756582.

[25] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey, "Extraction of visual features for lipreading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 198–213, Feb. 2002, doi: 10.1109/34.982900.

[26] A. Fernandez-Lopez and F. M. Sukno, "Lip-reading with limited-data network," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5, doi: 10.23919/EUSIPCO.2019.8902572.

[27] H. Wang, G. Pu, and T. Chen, "A lip reading method based on 3D convolutional vision transformer," *IEEE Access*, vol. 10, pp. 77205–77212, 2022, doi: 10.1109/ACCESS.2022.3193231.

[28] D. Feng, S. Yang, S. Shan, and X. Chen, "Learn an effective lip reading model without pains," 2020, *arXiv:2011.07557*.

[29] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *J. Acoust. Soc. Amer.*, vol. 120, no. 5, pp. 2421–2424, Nov. 2006, doi: 10.1121/1.2229005.

[30] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*. New York, NY, USA: ACM Press, 2006, pp. 369–376, doi: 10.1145/1143844.1143891.

[31] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3444–3450, doi: 10.1109/CVPR.2017.367.

[32] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4960–4964, doi: 10.1109/ICASSP.2016.7472621.

[33] S. Fenghour, D. Chen, K. Guo, and P. Xiao, "Lip reading sentences using deep learning with only visual cues," *IEEE Access*, vol. 8, pp. 215516–215530, 2020, doi: 10.1109/ACCESS.2020.3040906.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[35] A. M. Sarhan, N. M. Elshennawy, and D. M. Ibrahim, "HLR-Net: A hybrid lip-reading model based on deep convolutional neural networks," *Comput., Mater. Continua*, vol. 68, no. 2, pp. 1531–1549, 2021, doi: 10.32604/cmc.2021.016509.

[36] J. Peymanfard, M. R. Mohammadi, H. Zeinali, and N. Mozayani, "Lip reading using external viseme decoding," in *Proc. Int. Conf. Mach. Vis. Image Process. (MVIP)*, Feb. 2022, pp. 1–5, doi: 10.1109/MVIP53647.2022.9738749.

[37] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," 2015, *arXiv:1506.07503*.

[38] J. Tiedemann and L. Nygaard, "The OPUS corpus—Parallel and free," in *Proc. 4th Int. Conf. Lang. Resour. Eval. (LREC)*, 2004, pp. 1183–1186. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2004/pdf/320.pdf

[39] A. Fernandez-Lopez and F. M. Sukno, "End-to-end lip-reading without large-scale data," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 2076–2090, 2022, doi: 10.1109/TASLP.2022.3182274.

[40] A. Fernandez-Lopez, O. Martinez, and F. M. Sukno, "Towards estimating the upper bound of visual-speech recognition: The visual lip-reading feasibility database," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 208–215, doi: 10.1109/FG.2017.34.

[41] N. Harte and E. Gillen, "TCD-TIMIT: An audio-visual corpus of continuous speech," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 603–615, May 2015, doi: 10.1109/TMM.2015.2407694.

[42] M. Kim, H. Kim, and Y. M. Ro, "Speaker-adaptive lip reading with user-dependent padding," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 576–593, doi: 10.1007/978-3-031-20059-5_33.

[43] Z. Anvari and V. Athitsos, "A pipeline for automated face dataset creation from unlabeled images," in *Proc. 12th ACM Int. Conf. Pervasive Technol. Rel. Assistive Environ.*, Jun. 2019, pp. 227–235, doi: 10.1145/3316782.3321522.

[44] R. El-Bialy, D. Chen, S. Fenghour, W. Hussein, P. Xiao, O. H. Karam, and B. Li, "Developing phoneme-based lip-reading sentences system for silent speech recognition," *CAAI Trans. Intell. Technol.*, vol. 8, no. 1, pp. 129–138, Mar. 2023, doi: 10.1049/cit2.12131.

[45] P. Damien, N. Wakim, and M. Egea, "Phoneme-viseme mapping for modern, classical Arabic language," in *Proc. Int. Conf. Adv. Comput. Tools Eng. Appl.*, Jul. 2009, pp. 547–552, doi: 10.1109/actea.2009.5227875.

[46] A. Fernandez-Lopez and F. M. Sukno, "Survey on automatic lip-reading in the era of deep learning," *Image Vis. Comput.*, vol. 78, pp. 53–72, Oct. 2018, doi: 10.1016/j.imavis.2018.07.002.

[47] S. Fenghour, D. Chen, K. Guo, B. Li, and P. Xiao, "Deep learning-based automated lip-reading: A survey," *IEEE Access*, vol. 9, pp. 121184–121205, 2021, doi: 10.1109/ACCESS.2021.3107946.

[48] G. Sterpu, C. Saam, and N. Harte, "How to teach DNNs to pay attention to the visual modality in speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 1052–1064, 2020, doi: 10.1109/TASLP.2020.2980436.

[49] P. Damien, "Visual speech recognition of modern classic Arabic language," in *Proc. Int. Symp. Humanities, Sci. Eng. Res.*, Jun. 2011, pp. 50–55, doi: 10.1109/SHUSER.2011.6008499.

ZAMEN JABR received the bachelor's degree in computer science from the College of Science, University of Thi-Qar, Iraq, in 2005, and the master's degree in computer science from the College of Science, Basrah University, Iraq, in 2012. She is currently pursuing the Ph.D. degree in computer science with Iran University of Science and Technology. From 2012 to 2020, she was a Lecturer with Thi-Qar University. Her research interests include lip reading, deep learning, image processing, and artificial intelligence.

SAULEH ETEMADI received the Ph.D. degree from Michigan State University, in 2016. He is currently an Assistant Professor with the Computer Engineering Department, Iran University of Science and Technology (IUST), Tehran, Iran. Before joining IUST, he had a 13-year career at Microsoft Research as a Senior Research Software Development Engineer, where he worked on natural language processing and machine translation.

NASSER MOZAYANI received the B.Sc. degree in electrical engineering (computer hardware) from the Sharif University of Technology, Tehran, Iran, the M.Sc. degree in information systems from Supélec, Rennes, France, and the Ph.D. degree in informatics from the University of Rennes 1, Rennes, in 1998. He is currently an Associate Professor with the Computer Engineering Department, Iran University of Science and Technology, Tehran.