



Full length article

Read my lips: Artificial intelligence word-level arabic lipreading system

Waleed Dweik^{a,*}, Sundus Altorman^a, Safa Ashour^a^a University of Jordan, Queen Rania Street, Computer Engineering Dept., Amman 11942, Jordan

ARTICLE INFO

Article history:

Received 27 September 2021

Revised 4 March 2022

Accepted 10 June 2022

Available online 1 July 2022

Keywords:

Lipreading

Visual speech recognition

Deep neural networks

Convolution

Long short-term memory

ABSTRACT

Lipreading is the ability to recognize words or sentences from the mouth movements of a speaking person. This process is also known as Visual Speech Recognition (VSR). Lipreading has two main advantages: facilitate communication for people with hearing or speaking problems and aid speech recognition in noisy environments. In this paper, we propose a lipreading computing system capable of recognizing ten common Arabic words by performing word extraction from the mouth movements. The system receives a video of a person uttering an Arabic word as an input and outputs the text of the predicted word. During the implementation stage of the proposed system, three deep learning and neural network architectures are alternatively used to train, validate, and test the system using a locally collected and preprocessed dataset. The dataset contains 1051 videos and will be made available upon request. Moreover, a voting model that combines the three architectures is proposed. The highest testing accuracy (i.e. 82.84%) is achieved by leveraging the voting model.

© 2022 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Deep learning techniques extract useful features and patterns from data using the optimization of neural networks to solve real-world problems. Many fields such as natural language processing, computer vision, and speech recognition leverage deep learning to achieve tasks with performance better than human experts. The vast amount of data available in the big data era is one of the main causes behind the recent advances in deep learning techniques. As more data is available for training, validation, and testing purposes, deep learning is capable of accomplishing recognition tasks with much higher accuracy. Another important factor of the current deep learning breakthrough is the growth of the computing power with efficient resources and supercomputers.

The ongoing massive interest and research in the field of deep learning motivate engineers, programmers, mathematicians,

statisticians, etc to deploy these artificial intelligence (AI) techniques to handle repetitive and hard tasks conveniently and to solve many problems that could make life easier. Similarly, in this paper we propose to leverage advanced deep learning techniques to design a computing system called Read My Lips (RML) which is capable of accurately recognizing common Arabic words from mouth movements without any audio information.

The importance of lipreading (i.e. visual speech recognition (VSR)) in human communication and speech understanding was early highlighted by [1–4]. Millions of people around the world suffer from medical problems or disabilities that prevent them from generating sounds or/and hearing properly. For instance, some people lose the ability to generate clear sounds due to Vocal Cord Paralysis, Laryngeal Cancer, Spasmodic Dysphonia, or Dysarthria. On the other hand, according to World Health Organization (WHO) more than 5% of the world's population suffer from hearing problems [5]. The proposed system can greatly improve the communication capabilities for people with hearing or speaking disabilities, specifically in countries where Arabic is the official or co-official language.

Moreover, VSR is used to aid audio speech recognition (ASR) in noisy environments [6,7]. This combination is referred to as audio-visual speech recognition (AVSR) and can be leveraged to identify speech in multi-talker environments (e.g. recorded videos used in criminal investigations) and identify instructions given to systems or machines from humans in noisy environments such as underwater and space activities. Finally, VSR has a wide range of multi-

* Corresponding author.

E-mail address: w.dweik@ju.edu.jo (W. Dweik).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.



Production and hosting by Elsevier

media applications (e.g. surveillance and Internet telephony) and speech reconstruction systems [8].

The four main contributions of this work are listed below:

- I. Generating a dataset that consists of 1051 soundless videos from 73 native Arabic speakers (33 females and 40 males) uttering one of ten common Arabic words in different environments.
- II. Processing the dataset using a novel mouth frame extraction technique which takes into consideration the high frame rate and the color scheme (i.e. RGB vs. grayscale). The processed dataset will be available along with its description for any interested party upon request through email to the corresponding author.
- III. Investigating the utilization of three advanced deep learning models in the implementation of RML and deriving the models' parameters that guarantee the highest recognition accuracy when using the generated dataset to train, validate, and test the system.
- IV. Proposing a voting model that can be used to implement RML by combining the three investigated learning models and leveraging the strengths of each model.

The rest of this paper is organized as follows: Section 2 presents the most recent related work. Section 3 describes the design components and the implementation details of the RML system. The experimental setup and results are discussed in Section 4. Finally, conclusions and future work are presented in Section 5.

2. Related work

The authors in [9] presented a very recent survey on automated lipreading mechanisms and provided a comparison between the previously proposed solutions for the main stages of lipreading (i.e. feature extraction and classification networks). In this section, we focus on the most related work to our paper. Faubel et al. [10] combined audio-visual voice activity detection (VAD) with microphone array processing techniques to improve the speaker localization. Siatras et al. [11] utilized the increased deviation and values of the number of low intensity pixels in the mouth region of a speaking person as visual cues to aid speech detection. Pingxian et al. [12] proposed a lip detection system based on the computer vision library; OpenCV.

More recently, Kumar et al. targeted mitigating the problem of low prediction accuracies in VSR classification models when unseen data is used for testing [13]. This was achieved by deploying Generative Adversarial Networks (GANs) to generate new unseen data and use it to perform zero-shot learning which improved the accuracy of VSR systems by 27%. In [14], the authors modeled how visual speech is perceived and showed its use case on video compression. In addition, a view-temporal attention mechanism was proposed to model VSR while taking into consideration both the view dependence and the visemic importance. The experimental results showed absolute improvement of 5% in the viseme error rate.

Shah et al. investigated the linguistic and acoustic features of two state-of-the-art audio transformer models (i.e. wav2vec2.0 and Mockingjay) by probing each of their layers [15]. The experimental results showed that these models can significantly capture audio, fluency, suprasegmental pronunciation, syntactic and semantic text-based characteristics. For each category of characteristics, the authors identified a learning pattern and the best layer within a model to choose for feature extraction for downstream tasks. The audio-visual speech recognition models in [16] were modified in [17] to predict visemes given an input video. The updated models were used to perform experiments in bilin-

gual and monolingual settings in order to determine the occurrence of critical periods similar to those examined during perceptual attunement of infants. The experimental results using LRW and LRW1000 datasets showed strong correlations between theories in cognitive science, psychology and linguistics and deep neural based models on the effect of critical periods on language acquisition. The remaining related work can be divided into three subsections according to the granularity of the recognized speech:

2.1. Word-level recognition

Eldirawy and Ashour [18] built a system to lip-read the numbers from one to ten when spoken in the Arabic language. Three recognition methods were used: K-mean, fuzzy k-mean, and k-nearest neighbors (K-NN) classifiers with maximum recognition accuracy of 55.8%. Morade and Patnaik [19] proposed a lipreading algorithm based on localized active counter model (ACM) and a hidden Markov model (HMM). The algorithm was tested by recording videos of English digits from zero to nine from 16 speakers (8 male and 8 female) and the Cuave database. The recognition accuracy varied between 77.8% and 79.6%.

Alzahraa et al. [20] proposed a model that relies on lip movements to identify the words for the numbers in the range from zero to nine in the Arabic language. The model utilizes speeded up robust features (SURF), histogram of oriented gradient (HoG), and Haar techniques and the reported recognition accuracy is 96.2%. In [21], Wand et al. merged feed-forward and recurrent neural network layers into a single structure that was trained using back-propagating error gradients to recognize speech from soundless video recordings. The best reported word recognition accuracy is 79.6% using videos of 19 speakers and 51 words from the GRID corpus [22].

In [23], an end-to-end visual speech recognition system based on Long-Short Memory (LSTM) networks is presented by Petridis et al. The system contains two streams; one extracts features directly from the mouth and the other extracts features from the difference images. Each stream is modeled by an LSTM and the merging of the two streams takes place via a bidirectional LSTM (BiLSTM). The system was evaluated using OuluVS2 and CUAVE databases with recognition accuracy of 84.5%. Petridis et al. built on top of their latter system such that it takes multi-view images simultaneously to achieve 3.8% recognition accuracy improvement when three views (i.e. frontal, profile, and 45 degrees) are used [24]. Once again, Petridis et al. proposed an end-to-end audiovisual model based on residual networks and Bidirectional Gated Recurrent Units (BGRUs) [25]. The model extracts features directly from the image pixels and audio waveforms to perform word-level recognition. The proposed system provided a slight improvement of 0.3% over audio-only models using the Wild (LRW) database.

Kumar et al. presented a regression model to perform multi-view lipreading and generate an audio output [26]. The model consists of a view classifier followed by a Spatio-Temporal Convolutional Neural Network (STCNN) and BGRUs network. The OuluVS2 database was used to train and test the model and the Perceptual Evaluation of Speech Quality (PESQ) was used as the evaluation metric. The highest score of 2.315 was achieved with triple-view at 0, 45, and 60 degrees. In addition, the proposed model provided substantial delay improvements over previous work. Uttam et al. designed an encoder-decoder speaker-independent architecture which takes silent videos as input and outputs an audio spectrogram of the reconstructed speech [27]. The model consists of four components: pose classifier, decision network to select the right encoder, audio autoencoder, and video encoder that consists of seven three-dimensional convolutional layers followed by an LSTM layer. The proposed model was tested

using bilingual speech (i.e. English and Hindi) and the results were comparable with the ones in [26].

Weng and Kitani proposed a word-level visual lipreading method based on deep three-dimensional CNN with two streams (i.e. grayscale video and optical flow streams) and BiLSTM [28]. The method was tested on LRW dataset with accuracy of 84.11%. In [29], Shrivastava et al. proposed a novel end-to-end deep neural network model for word-level VSR in resource constrained environments such as mobile devices. Depthwise 3D convolution and channel shuffling were deployed to achieve 70% recognition accuracy using the LRW dataset with 6 times fewer parameters and 20 times smaller memory foot print.

One of the most related work to ours is the one presented by Elrefaei et al. [30]. The authors collected an Arabic Visual Speech Dataset (AVSD) that contains 1100 videos of 10 words that are used in daily communication between people. The words were uttered by 22 speakers (i.e. each speaker repeated every word 5 times) and the region of interest (ROI) was manually cropped from the video frames. The evaluation of AVSD was done using Support Vector Machine (SVM) model with 70% accuracy. In our dataset, 73 speakers are included which reduces the percentage of data duplication and the flipping augmentation technique is used to double the training and validation sets and generates a total of 1828 videos. Moreover, the ROI is extracted using a systematic accurate approach as described in subSection 3.2. Lastly, we propose three deep-learning techniques with a voting model to achieve 82.84% accuracy on unseen testing set.

Motivated by psychological studies which indicate that people do not fix their gaze at the lips region during a face-to-face conversations, Zhange et al. evaluated the effects of considering different facial regions (e.g. upper face, mouth, whole face, cheeks) with state of the art VSR models [31]. Moreover, the authors introduced an effective method based on Cutout to extract discriminative features for VSR which resulted in 85.02% and 45.24% accuracies for LRW and LRW-1000 datasets, respectively. Martinez et al. addressed the limitations in the word-level BGRU-based lipreading model by replacing the BRGU layers with multi-scale Temporal Convolutional Networks (TCN) [32]. In addition, the authors addressed the generalization issue of state-of-the-art methodologies by proposing a variable-length augmentation. The updated model was tested using LRW and LRW-1000 datasets and 85.3% and 41.4% accuracies were achieved, respectively. In [33], the authors proposed multiple innovations to bridge that gap between lipreading methodologies and its effective practical deployment: first, self-distillation was used to improve state of the art accuracy on LRW and LRW-1000 datasets to 88.5% and 46.6%, respectively. Second, Depthwise Separable Temporal Convolutional Network (DS-TCN) was used to reduce the computational cost. Third, knowledge distillation was used to recover the performance of lightweight models.

2.2. Phoneme/letter-level recognition

Matthews et al. [34] integrated speech cues from lip deformations to improve speech recognition in noisy environments. To extract features from lip images, the authors implemented three different methods: Active Shape Model (ASM), Active Appearance Model (AAM), and Multiscale Spatial Analysis (MSA) and the recognition accuracy were 27%, 42%, and 45% respectively. In [35], Damien proposed a novel viseme-based Arabic VSR approach by introducing a consonant–vowel detector and using two classifiers for the recognition of the “consonant part” and the “vowel part” of the phoneme. A dataset of 240 recordings, in which the lips of the speakers were marked in blue, was collected. The proposed approach recognizes one phoneme at a time and combines the recognized phonemes to recognize the word using a decision tree

with 81.67% accuracy. It is not clear from the paper how the dataset is divided into training and testing in order to guarantee that the testing part of the dataset is unseen. On the other hand, our proposed approach achieves 82.84% accuracy on unseen testing set using a dataset of 1828 recordings.

Al-Ghanim et al. [36] proposed “I See What You Say” (ISWYS); an Arabic speech recognition system. ISWYS uses motion estimation techniques to analyze videos and interpret lip movements into readable text one letter at a time which resulted in recognition accuracy of 70%. In [37], Koller et al. incorporated the outputs of deep convolutional neural network classifiers into a HMM approach for learning and modeling mouth shapes in the context of sign language recognition. The RWTH-PHOENIX-Weather corpus is used as a dataset [38]. For the vowel classes “A”, “O” and “U”, the model achieved above 60% precision. On the other hand, the model achieved the lowest precision of 28.6% for class “I” because of the confusion with the “E” class.

2.3. Sentence-level recognition

Sagheer et al. [39] proposed a lipreading system that combines a Hypercolumn Neural Network Model (HCM) with a five-state HMM. The efficacy of the system was evaluated for Arabic and Japanese languages. For Arabic language evaluation, nine sentences, each with three words, were gathered from eight different native Arabian speakers. The recognition accuracy for words and sentences were 79.5% and 62.9%, respectively. Motivated by the observation that lipreading performance improves for longer words, Assael et al. [40] proposed LipNet; an end-to-end sentence-level lipreading model for the English language. LipNet leverages spatiotemporal convolutional neural networks (STCNNs), Bidirectional Gated Recurrent Units (Bi-GRUs) and the connectionist temporal classification loss (CTC) to operate at the character-level and make sentence-level predictions. On the GRID corpus dataset, LipNet achieved 95.2% recognition accuracy.

In [41], Xu et al. presented LCArNet, an end-to-end lipreading system that deploys 3D CNN for feature extraction and BGRU network with CTC for backend decoding of characters to predict full phrases from the GRID corpus with 3% WER on seen test data. Shillingford et al. [42] designed Vision to Phoneme (V2P); a lipreading system that consists of three stages: a pipeline to convert a raw video into a sequence of lip videos (i.e. one phoneme per video), a deep neural network to map lip videos to a sequence of phoneme distributions, and a speech decoder to convert phonemes into sequence of words. A word error rate (WER) of 40.9% is achieved using a vast dataset with 3,886 h of general YouTube videos. Most recently, Margam et al. proposed an architecture to recognize ASCII characters and then predict spoken sentences from the GRID corpus with 8.6% WER on unseen test data. The architecture consists of 3D and 2D CNNs as frontend and BiLSTMs with CTC as backend.












3. RML design and implementation

The Arabic language has 28 letters and is the fifth most widely spoken language with 422 million speakers [43]. Nevertheless, few research work focused on Arabic language lipreading. RML is trained to recognize 10 common Arabic words listed in Table 1 along with English pronunciation and meaning. Mouth shapes are called visemes, which are the visual counterparts of phonemes. A single viseme may represent several phonemes. During speech, the mouth forms between 10 and 14 different shapes. Table 2 contains all Arabic letters mapped to their respective visemes and English phonemes[44].

Table 1
Arabic Words Recognized by RML.

Arabic Word	English Pronunciation	English Meaning
أسف	Aasef	Sorry
اليوم	Alyawm	Today
غداً	Ghadan	Tomorrow
جميل	Jameel	Beautiful
خير	Khair	Good
مرحباً	Marhaba	Hello
مساء	Masaa	Evening
صباح	Sabah	Morning
سلام	Salam	Peace (used in general greetings)
شكراً	Shukrun	Thanks

Table 2
Phoneme-Viseme Mapping for Arabic Language.

Arabic Character	English Phoneme	Viseme
ل، ن، ر	[r], [n], [l]	
ف	[f]	
و	[w]	
ط، د، ض، ت	[t], [dʃ], [d], [tʃ]	
ء، ق، ه، ع، ح	[h], [ʔ], [h], [q], [ʔ]	
ب، م	[m], [b/p]	
غ، خ، ك	[k], [χ], [ɣ]	
ي	[e(:)/j]	
ج، ش	[sh/ʃ], [ʒ]	
س، ز، ص	[sʃ], [z], [s]	
ظ، ث، ذ	[ð], [θ], [ðʕ]	

The design process of RML consists of three steps: dataset collection, dataset preprocessing and splitting, and prediction models implementation. The details of the three steps are described in the following three subsections, respectively. After the third step is completed, the preprocessed labeled dataset is used to train, validate, and test the implemented models as discussed in the [subSection 4.2](#). The implemented deep learning models perform

automatic feature extraction from the image frames which contain the lip movements and use the extracted features to classify the input video of the spoken word.

Once the models are trained, they can be used to classify an unseen input video as shown in the flowchart in [Fig. 1](#). The input video must be preprocessed using the steps listed in [subSection 3.2](#). The output of the preprocessing stage is 30 image frames of the lip

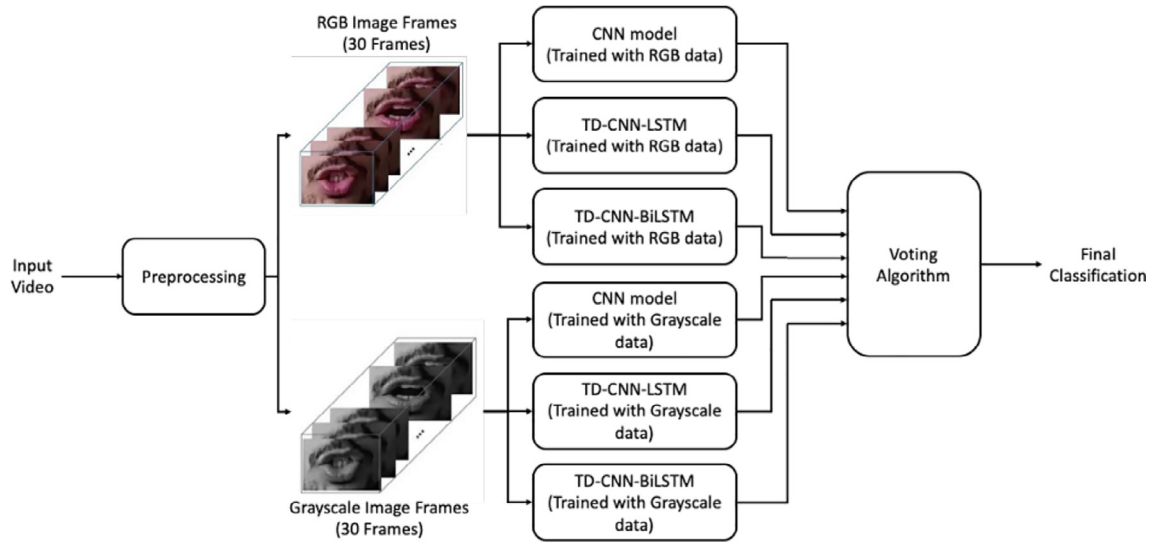


Fig. 1. RML Flowchart.

area in RGB and grayscale formats. Consequently, the frames are input to the three deep learning prediction models described in subSection 3.3. Given that each model is trained twice, once with the RGB version of the dataset and the other with the grayscale version of the dataset, a total of six models are deployed in RML. Each model will automatically extract features from the input frames and generate its prediction. Lastly, the predictions of the six models are processed by the voting algorithm explained in subSection 4.2.3 to generate the final classification of the input video.

3.1. Dataset collection

The input of the RML system is a soundless video, extracted to frames, of a person uttering one of the Arabic words in Table 1. The output of the system is the transcript of the recognized word. In order to implement and test the RML system, a sufficient dataset is required. Accordingly, 1051 videos were captured for different participants from both genders (i.e. 33 females and 40 males). All participants are university students (i.e. age range is 18-to-21 years) from the same country (i.e. one of the Levant area countries). The videos of the dataset are captured with different background lighting settings, different distances, and using cameras of multiple smartphones with the used smartphone placed in a holder. Different cameras, distances, and background lighting settings guarantee dataset generalization. All used cameras record videos at a rate of 30 frames per second (fps).

Most of the 73 speakers uttered each one of the 10 words once (i.e. 730 videos); however, few of them repeated the 10 words between two and four times (i.e. This accounts for the extra 321 videos). This ensures the inclusiveness of the dataset because there are many diversities between people that need to be accounted for, such as: speediness of speaking, mouth shape and movements, lips geometric features, amount of tongue determined by the redness of the taken mouth frame, the alveolar ridge, teeth, braces, mustache, beard, and makeup.

Two versions of the dataset are used in the experimental work: the colored version and the grayscale version. In the colored version, the video frames are represented by three channels: Red, Green, and Blue (i.e. RGB). The colored video frames are converted to grayscale frames to generate the second version of the dataset; where each pixel is represented by a single channel that reflects the brightness of the pixel.

3.2. Dataset preprocessing and splitting

One of the most important steps in deep learning tasks is to preprocess and split the dataset before feeding it to the learning algorithm. In RML, the following preprocessing steps are required:

- I. The lengths of all videos in the dataset are fixed to exactly one second while making sure that each video contains the target word without any errors. The one second period is chosen because most speakers require that period to utter one word. For few cases, when words are uttered in less than one second, the videos are trimmed and slow-motion is applied for the whole video or part of it.
- II. Each video is converted to 30 image frames using VideoCapture class from Python cv2 library because the videos are recorded at a rate of 30 fps.
- III. Each frame is processed using Python dlib face detector library which recognizes the frontal face of the person in the frame.
- IV. Each frame is processed to localize (i.e. extract) the lip region from the face. This is achieved by investigating the dlib landmark points of the face and finding out the best mouth cut that would be suitable for all people's mouths. As a result, the final frame would only contain the mouth region in order to isolate any interference or noise from the background or other parts of the face. For example, Fig. 2 includes a sample of the video frames that show the visemes for the word "Jameel".
- V. Each frame is resized to 66X100 pixels in order to reduce the dataset size without affecting the accuracy.
- VI. Each pixel value in a frame is normalized to a range between 0 and 1 without distorting differences between the frame's pixels. This is achieved by dividing each pixel value by 255.

After the last preprocessing step (i.e. normalization), the image frames are split into three sets: training, validation, and testing. The training set is used to train and fit the used deep neural network model and the validation set is used to tune the model hyperparameters and improve its performance. In addition, the validation set is used to avoid overfitting. The testing set is unseen data to evaluate the recognition accuracy of the used model. The dataset is split as follows: 652 videos in the training set, 125 videos in the validation set, and 274 videos in the testing set. A data augmenta-



Fig. 2. Video Frames for the Word "Jameel".

tion technique, namely flipping, is used to double the size of the training and the validation sets (i.e. 1304 videos for training and 250 videos for validation).

Lastly, the data is arranged in the correct order to enter the network. The dimensions for the RGB version of the data are $1828 \times 30 \times 66 \times 100 \times 3$; where 1828 is the total number of videos (including the flipped ones in the training and validation sets), 30 is the number of frames per video, 66×100 is the frame size in pixels, and the last dimension refers to the values of the three colors (i.e. red, green, and blue) for each pixel. For the grayscale version of the data, the dimensions are $1828 \times 30 \times 66 \times 100$ because each pixel has a single value. Furthermore, one-hot labels are assigned to the 10 data classes in Table 1 (i.e. Data labeling).

3.3. Prediction models

During the implementation phase of the RML, three deep learning models were investigated in order to choose the one with the highest recognition accuracy for the 10 Arabic words using the dataset described in subSection 3.1. In this subsection, we describe the details of the three models.

3.3.1. Convolutional neural networks

One of the main deep learning models used for recognition tasks is the Convolutional Neural Network (ConvNet or CNN) because it is capable of capturing spatial and temporal dependencies. CNN is an artificial neural network (ANN) that consists of an input layer followed by multiple hidden layers and then an output layer. The hidden layers contain one or more convolution layers normally followed by pooling, normalization, flatten, and fully-connected layers [45].

The Convolutional layers use kernels (i.e. filters) to extract features and recognize patterns in the input and generate what is called feature maps. A kernel is represented by a matrix that slides

along n-dimensions on the input data according to a specific stride. According to the value of n, a convolution layer can be classified to: 1-dimensional (Conv1D), 2-dimensional (Conv2D), and 3-dimensional (Conv3D). Notice that in order for the kernel to slide along n-dimensions of the input data, the shape of the input data should consist of at least $n + 1$ dimensions.

Fig. 3 shows the CNN model for the RML system when the RGB version of the dataset is used. The model contains two consecutive Conv3D layers with 32 kernels each, two consecutive Conv3D layers with 128 kernels each, and two consecutive Conv3D layers with 256 kernels each. The neurons in all these convolution layers use the rectified linear unit (ReLU) activation function. In this model, the shape of the RGB version of the dataset consists of four dimensions (i.e. frame number, frame height, frame width, and color channel) and the kernels slide along 3-dimensions of the input data; hence, Conv3D layers are used.

The CNN model in Fig. 3 includes a 3-dimensional MaxPooling layer after each two consecutive Conv3D layers. MaxPooling is one of the most frequently used pooling layers and it utilizes a max filter that slides along the layer's input according to its pool_size and stride. As the stride decreases, the number of extraction operations increases (i.e. more features are learned) and the dimensions of the layer's output increases. Notice that the max filter maintains the depth of the input data (i.e. no important features are lost).

The output of the last MaxPooling layer is unrolled to a 1-dimensional vector using a flatten layer. The vector is processed by fully connected layers (i.e. Dense layers) with dropout layers added between them. The dropout layers are used for regularization and to avoid overfitting. During training time, a dropout layer eliminates some of its inputs according to a specific rate and scales up the remaining inputs by $1/(1-\text{rate})$ such that the network never relies on a feature to be present.

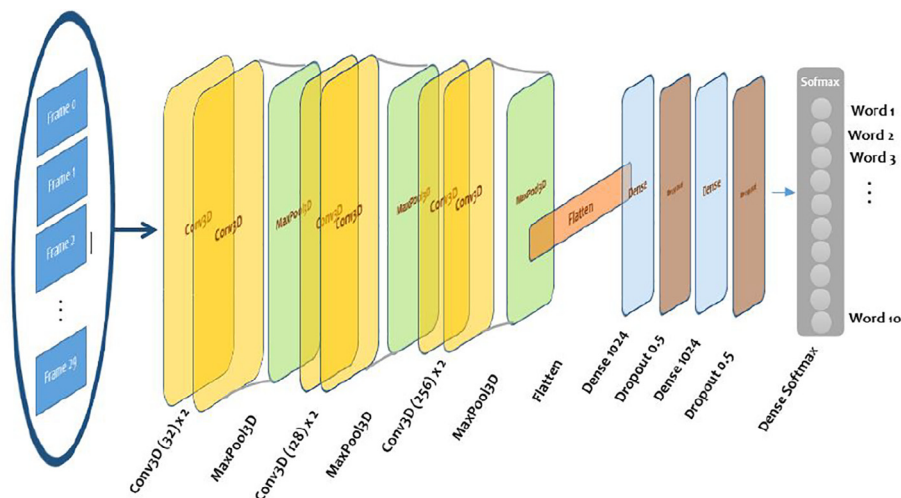


Fig. 3. RML CNN Model for RGB Dataset.

Finally, the output layer is a dense layer with 10 neurons that utilize Softmax activation function. Each neuron is associated with one of the 10 words or classes in Table 1 that the RML system is trained to recognize. The output of each neuron represents the probability that the word in the input video matches the associated class.

The numbers, sizes, and parameters of the layers in the CNN are chosen to achieve highest recognition accuracy. When the grayscale version of the dataset is used, the Conv3D and the 3-dimensional MaxPooling layers in the CNN model are replaced with Conv2D and 2-dimensional MaxPooling layers respectively. This is due to the fact that the shape of the grayscale version of the dataset consists of three dimensions (i.e. frame number, frame height, and frame width) and the kernels slide along two dimensions of the input data.

3.3.2. Time distributed convolutional neural network with long short-term memory

The structure of this model is to perform the feature extraction process through a time distributed (TD) convolution layer and then use a long short-term memory (LSTM) layer to process the extracted features and return sequences while keeping the sequence order. In the remaining of this paper, this model is referred to as TD-CNN-LSTM.

Fig. 4a shows the RML system when implemented using the TD-CNN-LSTM model and the RGB version of the dataset. The TD convolution layer works as a layer wrapper that allows the application of temporal parts of the input to separate layers which is useful when handling time-series data or video frames. In the RML system, each input video consists of 30 frames which are fed to 30 independent Conv2D layers with four kernels each. Since the frames are handled separately, the input shape of each convolution layer will consist of three dimensions (i.e. frame height, frame width, and color channel) and the kernels will slide along 2-dimensions.

As in the CNN model, each Conv2D layer in Fig. 4a is followed by a 2-dimensional MaxPooling layer and then a Flatten layer. Since no notable recognition accuracy improvements are noticed when adding more convolution layers (i.e. deep CNN), the TD-CNN-LSTM model is built using a single convolution layer per frame. When implementing the RML system using the TD-CNN-LSTM model and the grayscale version of the dataset, Conv1D and 1-dimensional MaxPooling layers are used.

LSTM is a recurrent neural network (RNN) architecture where layers interact in a way that provides the ability to remember information for long periods of time which is useful for the RML system. More design details about LSTM can be found in [46]. The outputs of the LSTM layer are fed to a dense layer with 256 neurons which are consequently connected to an output layer identical to that of the CNN model.

3.3.3. Time distributed convolutional neural network with bidirectional long short-term memory

This model shares the same front portion of the TD-CNN-LSTM model but the LSTM layer is doubled and made bidirectional; hence, in the remaining of this paper this model is referred to as TD-CNN-BiLSTM. Fig. 4b shows the structure of RML system when implemented using the TD-CNN-BiLSTM model and the RGB version of the dataset. The rationale for using a bidirectional layer is to avoid outweighing the output in the latest parts of the sequence frames.

4. Experimental setup and results

4.1. Experimental setup

The RML system is implemented in Python using Google Colaboratory (i.e. Colab) environment. The three prediction models discussed in Section 3.3 are coded using the Keras library that exists as a standalone but can also wrap multiple frameworks such as Tensorflow and Theano. All data and required files are stored in the Google drive associated with the Google account used to access the Colab environment.

When compiling the proposed prediction models, the Cross-Entropy (CE) Loss function and the Adam optimizer with learning rate (lr) of 0.0001 are used. The combination of the Softmax activation function utilized by the output layers of the proposed models and the CE Loss function is called Categorical CE Loss or Softmax Loss. The Categorical CE Loss function is commonly used when dealing with multi-class classification problems (i.e. more than two classes), where the classes are labeled using one-hot code and only one class can be correct. The objective is to minimize the Loss function of the prediction model. This happens when class with the highest prediction is the correct class.

Optimizers define how neural networks learn and find out the values that minimize the loss function. Adam optimizer is one of the most popular gradient descent advanced optimizers; it is computationally efficient and requires small amount of memory [47]. Table 3 shows the total number of parameters generated after compiling each predictive model per dataset version.

Table 3
Number of Parameters per Prediction Model.

Dataset Version	Prediction Model		
	CNN	TD-CNN-LSTM	TD-CNN-BiLSTM
Grayscale	3,240,874	463,806	2,369,726
RGB	18,132,298	6,754,170	14,951,546

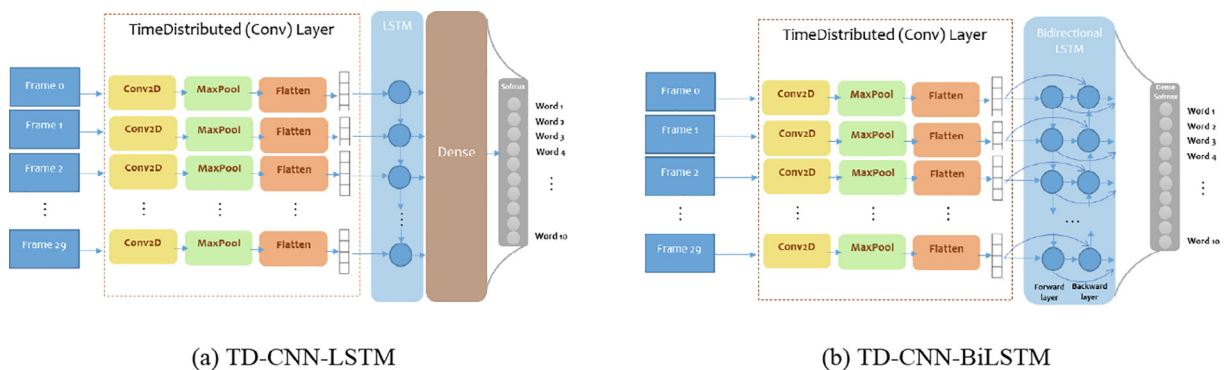


Fig. 4. RML TD-CNN-LSTM and TD-CNN-BiLSTM Models for RGB Dataset.

4.2. Simulation results

Given that the dataset splitting method described in [subSection 3.2](#) is performed randomly, the simulation results presented in the following three subsections are the average values of five simulation runs with five different random seeds for dataset splitting. In addition, the standard deviation for the testing accuracy of each model is reported.

4.2.1. CNN results

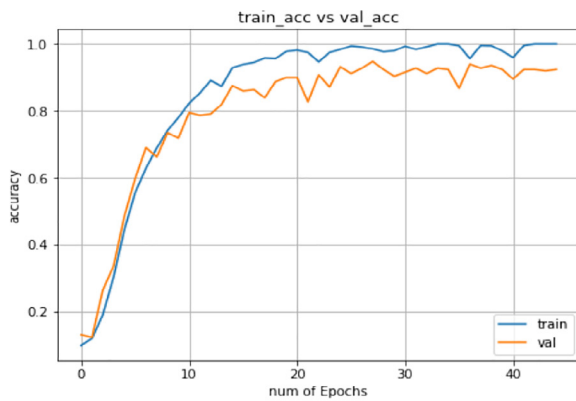
[Fig. 5](#) shows the prediction accuracy of the training and the validation sets when implementing the RML system using the CNN model with grayscale and RGB dataset versions. For the grayscale version, 45 epochs are required for the training and validation accuracy to saturate at 100% and 92.3% respectively. On the other hand, the training and validation accuracy of the RGB version saturate at 100% and 89.5% respectively after 30 epochs.

Alternatively, [Fig. 6](#) shows the loss function of the training and the validation sets when implementing the RML system using the CNN model with grayscale and RGB dataset versions. Generally, the value of the loss function decreases with more epochs. For the training sets of the grayscale and the RGB versions, the loss function eventually reaches the optimal value of zero. For the validation sets of the grayscale and the RGB versions, the loss function

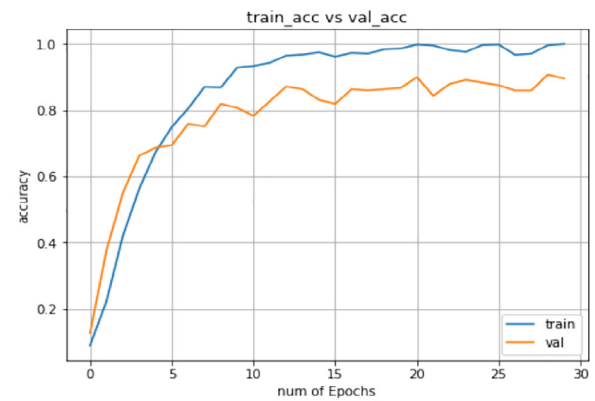
reaches 0.35 and 0.5 respectively. As with the accuracy, once the loss function starts to saturate the training and the validation processes are stopped to avoid overfitting.

[Fig. 7](#) shows the confusion matrices for the CNN model with grayscale and RGB versions of the testing set. The sum of the numbers in each row represents the number of samples of the corresponding class in the testing set. For example, the top most row in the confusion matrix of the grayscale version indicates that the testing set contains 35 samples of class Aasef out of which 27 are correctly predicted (i.e. true positives(tp)) and eight samples are mispredicted (i.e. false positives(fp)). On the other hand, by considering each column in the confusion matrix one can compute the number of false negatives (fn) of the corresponding class. For example, the leftmost column in the confusion matrix of the grayscale version indicates that the number of false negatives of class Aasef is 6.

[Tables 4 and 5](#) contain the precision (i.e. $tp/(tp + fp)$) and recall (i.e. $tp/(tp + fn)$) of each class and the total prediction accuracy of the grayscale and the RGB versions of the testing set respectively. It can be noticed that the words Ghadan and Khair are the hardest to be correctly predicted due to the fact that these words are short and the mouth movements of the characters of each word are very similar. The second observation is that the CNN model achieves higher overall prediction accuracy when RGB version of the dataset



(a) Grayscale

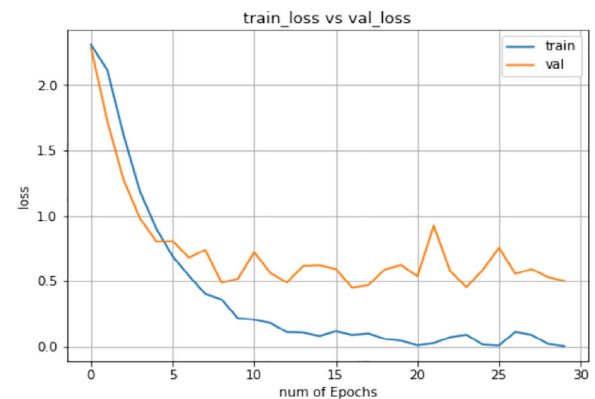


(b) RGB

Fig. 5. Training Accuracy vs Validation Accuracy for CNN Model.



(a) Grayscale



(b) RGB

Fig. 6. Loss Function for CNN Model.

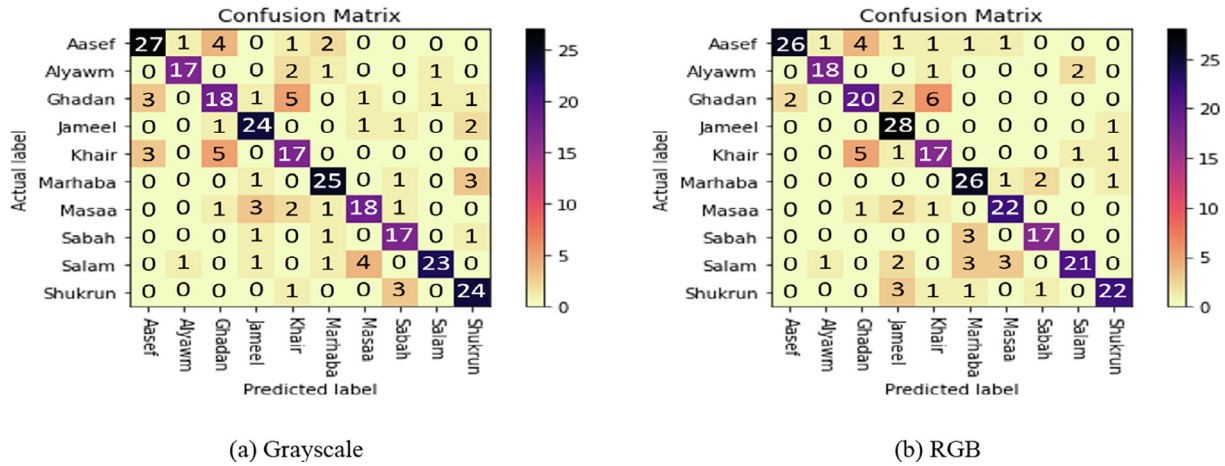


Fig. 7. Confusion Matrix for CNN Model.

Table 4

Precision, Recall, and Testing Accuracy for CNN Model with Grayscale Dataset.

Class	Aasef	Alyawm	Ghadan	Jameel	Khair	Marhaba	Masaa	Sabah	Salam	Shukrun
Precision	0.77	0.81	0.6	0.83	0.68	0.83	0.69	0.85	0.77	0.86
Recall	0.82	0.89	0.62	0.77	0.61	0.81	0.75	0.74	0.92	0.77
Total Prediction Accuracy of Testing Set = 0.766 ± 1.3										

Table 5

Precision, Recall, and Testing Accuracy for CNN Model with RGB Dataset.

Class	Aasef	Alyawm	Ghadan	Jameel	Khair	Marhaba	Masaa	Sabah	Salam	Shukrun
Precision	0.74	0.86	0.67	0.97	0.68	0.87	0.85	0.85	0.7	0.79
Recall	0.93	0.9	0.67	0.72	0.63	0.76	0.81	0.85	0.88	0.88
Total Prediction Accuracy of Testing Set = 0.792 ± 1.15										

is used (i.e. $0.792 > 0.766$). This indicates that when the image frames of the videos contain colors such as red for the tongue and white for the teeth, the CNN model is capable of extracting precise mouth movement features from the training set. These features are then used to categorize the testing set with high accuracy.

4.2.2. TD-CNN-LSTM and TD-CNN-BiLSTM results

The behaviors of the prediction accuracy and loss functions of the training and the validation sets when implementing the RML system using the TD-CNN-LSTM model or the TD-CNN-BiLSTM model are very similar to those of the CNN model. The main difference is that 80 epochs are required to saturate the prediction accuracy and loss functions of the TD-CNN-LSTM model. On the other hand, 40 and 45 epochs are required to saturate the prediction accuracy and loss functions of the TD-CNN-BiLSTM model with grayscale and RGB datasets, respectively.

The precision and recall of each class and the total prediction accuracy of the testing set for the TD-CNN-LSTM model with grayscale and RGB datasets are given in Tables 6 and 7, respectively. Similarly, Tables 8 and 9 contain the precision and recall of each class and the total prediction accuracy of the testing set for the TD-CNN-BiLSTM model with grayscale and RGB datasets, respectively. The four tables show that the general behavior of the TD-CNN-LSTM and TD-CNN-BiLSTM models coincide with that of the CNN model, as the words Ghadan and Khair are the hardest to be recognized and the overall prediction accuracy of the RGB dataset is higher than the grayscale dataset.

4.2.3. Voting model

The overall prediction accuracy of the three models indicate that CNN has the highest recognition accuracy followed by TD-CNN-BiLSTM then TD-CNN-LSTM. The order stays the same irrespective to the dataset version (i.e. grayscale or RGB) used. The CNN model applies convolution in 2-dimensions and 3-dimensions for the grayscale and RGB versions of the dataset respectively. On the other hand, the TD-CNN-LSTM and the TD-CNN-BiLSTM models apply convolution in 1-dimension and 2-dimensions for the grayscale and RGB versions of the dataset respectively. As a result, the CNN model generates more trainable parameters as shown in Table 3 which improves the prediction accuracy.

The deployment of the bidirectional layer in the TD-CNN-BiLSTM model allows it to extract more features than the TD-CNN-LSTM model as shown in Table 3. Moreover, the bidirectional layer helps the model to learn the features better and faster because the model learns from past and future at the same time. Hence, overall the TD-CNN-BiLSTM model performs better than the TD-CNN-LSTM model.

Despite that the CNN model has the highest overall prediction accuracy, the prediction accuracy of a specific class might be higher when using the TD-CNN-LSTM model or the TD-CNN-BiLSTM model. For example, the precision values of class Sabah in Tables 4, 6, and 8 show that the CNN model correctly predicts class Sabah from the grayscale dataset 85% of the time. On the other hand, the TD-CNN-LSTM and the TD-CNN-BiLSTM models correctly predict class Sabah from the grayscale dataset 90% and 95% of the time, respectively.

Table 6

Precision, Recall, and Testing Accuracy for TD-CNN-LSTM Model with Grayscale Dataset.

Class	Aasef	Alyawm	Ghadan	Jameel	Khair	Marhaba	Masaa	Sabah	Salam	Shukrun
Precision	0.6	0.52	0.4	0.83	0.4	0.83	0.73	0.9	0.73	0.82
Recall	0.62	0.69	0.46	0.75	0.43	0.74	0.7	0.69	0.81	0.82
Total Prediction Accuracy of Testing Set = 0.675 ± 1.3										

Table 7

Precision, Recall, and Testing Accuracy for TD-CNN-LSTM Model with RGB Dataset.

Class	Aasef	Alyawm	Ghadan	Jameel	Khair	Marhaba	Masaa	Sabah	Salam	Shukrun
Precision	0.6	0.76	0.57	0.77	0.56	0.7	0.62	0.85	0.83	0.82
Recall	0.75	0.67	0.57	0.73	0.45	0.68	0.76	0.89	0.78	0.82
Total Prediction Accuracy of Testing Set = 0.701 ± 0.75										

Table 8

Precision, Recall, and Testing Accuracy for TD-CNN-BiLSTM Model with Grayscale Dataset.

Class	Aasef	Alyawm	Ghadan	Jameel	Khair	Marhaba	Masaa	Sabah	Salam	Shukrun
Precision	0.63	0.76	0.47	0.9	0.44	0.87	0.65	0.95	0.63	0.79
Recall	0.67	0.84	0.63	0.81	0.31	0.76	0.74	0.79	0.86	0.76
Total Prediction Accuracy of Testing Set = 0.701 ± 0.75										

Table 9

Precision, Recall, and Testing Accuracy for TD-CNN-BiLSTM Model with RGB Dataset.

Class	Aasef	Alyawm	Ghadan	Jameel	Khair	Marhaba	Masaa	Sabah	Salam	Shukrun
Precision	0.71	0.76	0.53	0.86	0.6	0.63	0.81	0.95	0.8	0.82
Recall	0.76	0.84	0.55	0.86	0.41	0.9	0.78	0.86	0.83	0.82
Total Prediction Accuracy of Testing Set = 0.741 ± 1.77										

Likewise, despite that the TD-CNN-BiLSTM model has higher overall prediction accuracy than the TD-CNN-LSTM model, the prediction accuracy of a specific class might be higher when using the TD-CNN-LSTM model. For example, the precision values of class Salam in [Tables 5, 7, and 9](#) show that the CNN and TD-CNN-BiLSTM models correctly predict class Salam from the RGB dataset 70% and 80%, respectively. On the other hand, the TD-CNN-LSTM model correctly predicts class Salam from the RGB dataset 83% of the time.

Although it is observed that for a specific model the overall prediction accuracy for the RGB dataset is higher than the grayscale dataset, for a specific class the opposite might be true. For example, the precision values of classes Aasef, Salam, and Shukrun in [Table 4](#) (i.e. CNN model with grayscale dataset) are higher than those in [Table 5](#) (i.e. CNN model with RGB dataset).

Based on the observations in the previous three paragraphs, a voting model which simultaneously runs six prediction models (i.e. two instances of each model, one instance with the grayscale dataset and one instance with the RGB dataset) is proposed. Each input video, after being preprocessed, is fed to the six instances and the final prediction of each instance is considered as a vote. When there is a majority vote, it is chosen as the final output of the voting model. There are three scenarios where a majority vote exists: four or more identical votes, three identical votes and the

remaining three are not identical, or two identical votes and each vote of the remaining four is unique.

When there is no majority vote, the vote with the highest occurrence frequency is chosen. If more than one vote have the same occurrence frequency, the average prediction accuracy is used as a tie breaker. For example, let the six votes be Salam, Salam, Salam, Sabah, Sabah, Sabah and their prediction accuracy are 0.9, 0.9, 0.6, 0.8, 0.8, and 0.5 respectively. The occurrence frequencies of votes Salam and Sabah equal three but the average prediction accuracy of votes Salam and Sabah are 0.8 and 0.7, respectively. Hence, class Salam is the final output. Another example is if the six votes are Aasef, Salam, Khair, Khair, Jameel, and Alyawm. Regardless of the prediction accuracy of the votes, class Khair is the final output because it has the highest occurrence frequency. In the unlikely scenario of multiple votes with equal occurrence frequency and equal average prediction accuracy, the final output is chosen randomly.

The precision and recall of each class and the total prediction accuracy for the voting model on the testing set are given in [Table 10](#). The overall prediction accuracy of the voting model is 0.8284 which is 4.6% higher than the best overall prediction accuracy achieved by the CNN model with the RGB dataset. Moreover, the voting algorithm improves the precision values of the four classes Marhaba, Masaa, Salam, and Shukran to 0.9, 0.88, 0.9, and

Table 10

Precision, Recall, and Testing Accuracy for Voting Model

Class	Aasef	Alyawm	Ghadan	Jameel	Khair	Marhaba	Masaa	Sabah	Salam	Shukrun
Precision	0.77	0.81	0.6	0.97	0.64	0.9	0.88	0.95	0.9	0.89
Recall	0.79	0.85	0.64	0.93	0.55	0.9	0.88	0.95	0.93	0.89
Total Prediction Accuracy of Testing Set = 0.8284 ± 1.5										

0.89 respectively. These precision values are higher than the previous best precision values (i.e. 0.87 for Marhaba in Tables 5 and 8, 0.85 for Masaa in Table 5, 0.83 for Salam in Table 7, and 0.86 for Sukran in Table 4). For the three classes Aasef, Jameel, and Sabah, the voting model maintains the previous best precision values. On the other hand, for the classes Alyawm, Ghadan, and Khair the voting model achieves precision values of 0.81, 0.6, and 0.64 respectively. The latter precision values are less than the previous best precision values (i.e. 0.86 in Table 5, 0.67 in Table 5, and 0.68 in Tables 4 and 5).

5. Conclusions and future work

In this paper, we introduce Read My Lips (RML); a word-level Arabic lipreading system. Three deep learning models were investigated during and implementation phase of RML: CNN, TD-CNN-LSTM, and TD-CNN-BiLSTM. In order to train, validate, and test the system, two datasets which contain grayscale and RGB video frames of 73 speakers uttering ten common Arabic words was collected. The experimental results show that the CNN model with RGB dataset has the highest overall prediction accuracy of 79.2% on unseen test data. In addition, a voting model is proposed to implement the RML system which improves the overall prediction accuracy to 82.84%.

We have two future directions in mind: first, we plan to increase the size of the dataset by including more Arabic words and more videos of the same word. The purpose of this extension is to validate our current results for a larger dataset and improve the prediction accuracy of the models. Second, we plan to extend RML to become a sentence-level Arabic lipreading system so that it can be used in real-time applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

aaa

References

- [1] Woodward MF, Barber CG. Phoneme perception in lipreading. *J Speech Hear Res* 1960;3:212–22.
- [2] Fisher CG. Confusions among visually perceived consonants. *J Speech Hear Res* 1968;11:796–804.
- [3] McGurk H, Macdonald J. Hearing lips and seeing voices. *Nature* 1976;264:746–8.
- [4] Easton RD, Basala M. Perceptual dominance during lipreading. *Perception Psychophys* 1982;32:562–70.
- [5] Deafness. URL: <https://www.who.int/news-room/facts-in-pictures/detail/deafness>, accessed: 2020-04-15..
- [6] Goldschen AJ, Garcia ON, Petajan ED. Continuous automatic speech recognition by lipreading. In: *Motion-Based Recognition*. Dordrecht: Springer Netherlands; 1997. p. 321–43.
- [7] Potamianos G, Neti C, Gravier G, Garg A, Senior AW. Recent advances in the automatic recognition of audiovisual speech. *Proc IEEE* 2003;91(9):1306–26.
- [8] Kumar Y, Aggarwal M, Nawal P, Satoh S, Shah RR, Zimmermann R. Harnessing ai for speech reconstruction using multi-view silent video feed. In: *Proceedings of the 26th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery; 2018. p. 1976–83.
- [9] Fenghour S, Chen D, Guo K, Li B, Xiao P. Deep learning-based automated lipreading: A survey. *IEEE Access* 2021;9:121184–205.
- [10] Faubel F, Georges M, Kumatan K, Bruhn A, Klakow D. Improving hands-free speech recognition in a car through audio-visual voice activity detection. *Joint Workshop on Hands-free Speech Communication and Microphone Arrays* 2011;2011:70–5.
- [11] Siatras S, Nikolaidis N, Pitas I. Visual speech detection using mouth region intensities. *14th European Signal Processing Conference* 2006:1–5.
- [12] Pingxian Y, Rong G, Peng G, Zhaoju F. Research on lip detection based on opencv. In: *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*. p. 1465–8.
- [13] Kumar Y, Sahrawat D, Maheshwari S, Mahata D, Stent A, Yin Y, Ratn Shah R, Zimmermann R. Harnessing gans for zero-shot learning of new classes in visual speech recognition. *Proceedings of the AAAI Conference on Artificial Intelligence* 2020;34 (03):2645–2652..
- [14] Sahrawat D, Kumar Y, Aggarwal S, Yin Y, Shah RR, Zimmermann R. Notic my speech – blending speech patterns with multimedia; 2020. arXiv:2006.08599..
- [15] Shah J, Singla YK, Chen C, Shah RR. What all do audio transformer models hear? probing acoustic representations for language delivery and its structure; 2021. arXiv:2101.00387..
- [16] Petridis S, Stafylakis T, Ma P, Tzimiropoulos G, Pantic M. Audio-visual speech recognition with a hybrid ctc/attention architecture. *IEEE Spoken Language Technology Workshop (SLT)* 2018;2018:513–20.
- [17] Saraswat A, Bhatia M, Singla YK, Chen C, Shah RR. Perception point: Identifying critical learning periods in speech for bilingual networks; 2021. arXiv:2110.06507..
- [18] Eldirawy I, Ashour W. Visual speech recognition: The digits from one to ten in the arabic language. LAP LAMBERT Academic Publishing; 2011.
- [19] Morade SS, Patnaik S. A novel lip reading algorithm by using localized acm and hmm: Tested for digit recognition. *Optik* 2014;125(18):5181–6.
- [20] Reda AH, Nasr AA, Ezz MM, Harb HM. An arabic figures recognition model based on automatic learning of lip movement. *J Al Azhar Univ Eng Sector* 2017;12(42):155–65.
- [21] Wand M, Koutník J, Schmidhuber J. Lipreading with long short-term memory; 2016. arXiv:1601.08188..
- [22] Cooke M, Barker J, Cunningham S, Shao X. An audio-visual corpus for speech perception and automatic speech recognition. *J Acoust Soc Am* 2006;120 (5):2421–4.
- [23] Petridis S, Li Z, Pantic M. End-to-end visual speech recognition with lstms. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. p. 2592–6.
- [24] Petridis S, Wang Y, Li Z, Pantic M. End-to-end multi-view lipreading; 2017. arXiv:1709.00443..
- [25] Petridis S, Stafylakis T, Ma P, Cai F, Tzimiropoulos G, Pantic M. End-to-end audiovisual speech recognition. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. p. 6548–52.
- [26] Kumar Y, Jain R, Salik KM, Shah RR, Yin Y, Zimmermann R. Lipper: Synthesizing thy speech using multi-view lipreading; 2019. arXiv:1907.01367..
- [27] Uttam S, Kumar Y, Sahrawat D, Aggarwal M, Shah RR, Mahata D, Stent A. Hush-Hush Speak: Speech Reconstruction Using Silent Videos. In: *Proc. Interspeech* 2019; 2019. pp. 136–140..
- [28] Weng X, Kitani K. Learning spatio-temporal features with two-stream deep 3d cnns for lipreading; 2019. arXiv:1905.02540..
- [29] Shrivastava N, Saxena A, Kumar Y, Shah RR, Stent A, Mahata D, Kaur P, Zimmermann R. MobiVSR: Efficient and Light-Weight Neural Network for Visual Speech Recognition on Mobile Devices. In: *Proc. Interspeech* 2019; 2019. pp. 2753–2757..
- [30] Elrefaei LA, Alhassan TQ, Omar SS. An arabic visual dataset for visual speech recognition. *Proc Comput Sci* 2019;163:400–9.
- [31] Zhang Y, Yang S, Xiao J, Shan S, Chen X. Can we read speech beyond the lips? rethinking roi selection for deep visual speech recognition; 2020. arXiv:2003.03206..
- [32] Martinez B, Ma P, Petridis S, Pantic M. Lipreading using temporal convolutional networks. In: *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. p. 6319–23.
- [33] Ma P, Martinez B, Petridis S, Pantic M. Towards practical lipreading with distilled and efficient models. In: *ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. p. 7608–12.
- [34] Matthews I, Cootes TF, Bangham JA, Cox S, Harvey R. Extraction of visual features for lipreading. *IEEE Trans Pattern Anal Mach Intell* 2002;24 (2):198–213.
- [35] Damien P. Visual speech recognition of modern classic arabic language, in: *2011 International Symposium on Humanities*. Sci Eng Res 2011;2011:50–5.
- [36] Al-Ghanim A, Al-Oboud N, Al-Haidary R, Al-Zeer S, Altammami S, Mahmoud HA. I see what you say (iswys): Arabic lip reading system. In: *2013 International Conference on Current Trends in Information Technology (CTIT)*; 2013. pp. 11–17..
- [37] Koller O, Ney H, Bowden R. Deep learning of mouth shapes for sign language. *IEEE International Conference on Computer Vision Workshop (ICCVW)* 2015;2015:477–83.
- [38] Forster J, Schmidt C, Hoyoux T, Koller O, Zelle U, Piater J, Ney H. Rwth-phoenix-weather: A large vocabulary sign language recognition and translation corpus. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey; 2012. p. 3785–9.
- [39] Sagheer A, Tsuruta N, Taniguchi R. Arabic lip-reading system: A combination of hypercolumn neural network model with hidden markov model. In: *The Eighth IASTED International Conference on Artificial Intelligence and Soft Computing*. p. 311–6.
- [40] Assael YM, Shillingford B, Whiteson S, de Freitas N. Lipnet: End-to-end sentence-level lipreading; 2016. arXiv:1611.01599..
- [41] Xu K, Li D, Cassimatis N, Wang X. Lcanet: End-to-end lipreading with cascaded attention-ctc. In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. p. 548–55.

- [42] Shillingford B, Assael Y, Hoffman MW, Paine T, Hughes C, Prabhu U, Liao H, Sak H, Rao K, Bennett L, Mulville M, Coppin B, Laurie B, Senior A, de Freitas N. Large-scale visual speech recognition; 2018. arXiv:1807.05162..
- [43] List of countries where arabic is an official language, URL: https://en.wikipedia.org/wiki/List_of_countries_where_Arabic_is_an_official_language, accessed: 2020-04-15..
- [44] Damien P, Wakim N, Egea M. Phoneme-viseme mapping for modern, classical arabic language. International Conference on Advances in Computational Tools for Engineering Applications 2009;2009:547–52.
- [45] Khan A, Sohail A, Zahoora U, Saeed A. A survey of the recent architectures of deep convolutional neural networks. *Artif Intell Rev* 2020;53:5455–516.
- [46] Yu Y, Si X, Hu C, Zhang J. A review of recurrent neural networks: Lstm cells and network architectures. *Neural Comput* 2019;31:1–36.
- [47] Kingma DP, Ba J. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, May 7–9, 2015..