International Journal of Speech Technology
https://doi.org/10.1007/s10772-021-09956-3

A comparison of neural-based visual recognisers for speech activity detection

Sajjadali Raza[1] · **Heriberto Cuayáhuitl**[1]

Received: 30 July 2020 / Accepted: 24 December 2021

Abstract
Existing literature on speech activity detection (SAD) highlights different approaches within neural networks but does not provide a comprehensive comparison to these methods. This is important because such neural approaches often require hardware-intensive resources. In this article, we provide a comparative analysis of three different approaches: classification with still images (CNN model), classification based on previous images (CRNN model), and classification of sequences of images (Seq2Seq model). Our experimental results using the Vid-TIMIT dataset show that the CNN model can achieve an accuracy of 97% whereas the CRNN and Seq2Seq models increase the classification to 99%. Further experiments show that the CRNN model is almost as accurate as the Seq2Seq model (99.1% vs. 99.6% of classification accuracy, respectively) but 57% faster to train (326 vs. 761 secs. per epoch).

Keywords Visual speech activity recognition · Convolutional neural networks · Recurrent neural networks

1 Introduction

The task of detecting speech is typically referred to as voice activity detection (VAD) or speech activity detection (SAD) in the existing literature. SAD can be considered as classifying a video frame or image as speech or non-speech. Traditional approaches have often used audio signals for SAD but in the recent literature, approaches either involve the use of video signals or a combination of both (audio and video). In noisy environments, non-speech can often be classified as speech due to an increase in noise (Le Cornu and Milner 2015; Ariav and Cohen 2019; Sharma et al. 2019). As a consequence, recent deep learning approaches applied to SAD use video/images or a combination of audio and video for increased robustness.

In recent years, modern deep neural networks (DNNs)—especially applied to SAD—rely on recurrent neural networks (RNNs) due to their ability to learn temporal dynamic behaviour (Ariav and Cohen 2019; Sharma et al. 2019). Recent literature shows that authors have opted for a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). In this case, CNNs are used to learn image representations as they carry the ability to distinguish between images. But the addition of RNNs allows the network to learn temporal information, as it introduces history and information regarding the sequence Sharma et al. (2019).

In this article, we compare three different approaches in the literature for detecting whether a person is speaking or not: classification based on still images (one-to-one); classification based on previous images (many-to-one); and sequence classification (many-to-many). Our results show that

1. Using a history of images is crucial for improved performance,
2. The many-to-one approach is marginally outperformed by the many-to-many approach, and
3. The many-to-one approach requires much less computional requirements than the many-to-many approach and represents the best compromise across performance metrics.

✉ Heriberto Cuayáhuitl
h.cuayahuitl@lincoln.ac.uk

Sajjadali Raza
sajjadaliraza@live.co.uk

[1] Brayford Pool, Lincoln, UK

Published online: 08 June 2022

2 Literature review

Le Cornu and Milner (2015) proposed a technique to extract visual speech features and use them to classify between speech and non-speech. They compare the performances of CNNs (convolutional neural networks) and GMMs (Gaussian mixture models). Their suggested architecture consists of 2 convolution layers (3 x 3 x 32, 3 x 3 x 64), followed by max-pooling (2 x 2) at each convolutional layer with a dropout of 0.2 for convolutional layers. The RELU-based architecture also includes L2 regularization at 0.0001 and a dropout of 0.5 in the fully-connected layer with 512 neurons. GRID Cooke et al. (2006) is the dataset used in their experiments in two different scenarios. In the first scenario, data is split based on the ratio of 80:20 for training and testing respectively (speaker-independent). In the second scenario, data is split based on the speaker at the ratio of 80:20 (speaker-dependent). For speaker-dependent, the CNN achieves an accuracy of 97.66% and the GMM 94.34%. For speaker-independent, the CNN achieves a classification accuracy of 74.68% whereas the GMM achieves only 70.50%. Le Cornu and Milner (2015) also explore temporal information in the CNN whereby the first and last frames of the sequence are included. However, this only resulted in a slight increase in classification accuracy. As a result, they suggest further exploration with different architectures regarding temporal information.

Sharma et al. (2019) extends the work of Le Cornu and Milner (2015) by exploring temporal information as suggested and thus combine a CNN with an RNN. The paper involves visual SAD but is focused on the endpoint—when one stops speaking. Their architecture involves a 3-layered CNN with 5 x 5 filters with kernel sizes of 16, 32, and 8 respectively with a stride of 2. For every layer of the CNN, max-pooling(2 x 2) is added as well as a batch normalization layer. Two unidirectional Long-Short Term Memory networks (LSTMs) are added with a hidden size of 64. The state is then passed to a dense layer followed by a Softmax layer to determine the frame as speech or non-speech. Classification of the endpoint is based on the sequence, consisting of classification for each frame. Their network is run on multiple datasets such as GRID, VidTIMIT Sanderson and Lovell (2009) and a personally collected dataset referred to as "Indian-English dataset". Similar to Le Cornu and Milner (2015), their experiments are run on speaker-dependent and speaker-independent scenarios. They compare the performance of Le Cornu and Milner (2015) and their suggested architecture on the GRID dataset. Their results show that there is an increase in performance with the introduction of the RNN. Their architecture achieves an accuracy of 92.2% in the speaker-independent scenario as opposed to Le Cornu and Milner (2015) achieving only 74.68%. For speaker-dependent, the accuracy achieved is 96.5% which is a similar result achieved by Le Cornu and Milner (2015).

Wang and Wang (2019) introduced the landmark pooling network (LPN), which acts as an attention guide scheme to help the network only focus on the region of interest (ROI). The LPN network is provided with raw images including landmarks to focus on. This eliminates any pre-processing and computation to obtain ROI images and reduces the size of the network. Furthermore, allows assigning higher values to important locations. LPN uses a convolution layer (7 x 7 x 64), followed by a landmark pooling layer to pull the feature maps. The landmark pooling layer uses 20 landmarks around the mouth and thus has 20 64-dimensional vectors, which are passed to a Fully Connected (FC) Layer. Such FC layer is then passed to a GRU (gated recurrent unit) with 64 hidden units, at which case classification (AdaGrad with a learning rate of 0.0001) is made via a Softmax layer.

The dataset for their network is personally collected and referred to as "Labelled Speech in the Wild (LSW)". The dataset consists of speech and non-speech sequences with 195 subjects and 8903 sequences. 171 subjects and 8002 sequences are used for training and 24 subjects with 901 sequences are used for testing. Data is augmented using random flipping, cropping, and face movement speed to name a few. Their highest accuracy for their network is 79.9% with a network that involves LPN and CNN—using CNN features of the image and landmarks as input. The LPN network alone achieves 72.1% compared to a CNN (one convolutional layer, spatial max pooling layer, GRU) achieving a classification accuracy of 76.7%.

Whilst most previous works focus on recognizing speech activity using images of the mouth region, other alternatives include whole face images and whole upper body images. The study of Joosten et al. (2015) compared whole face images and mouth region found that the latter achieves higher classification results in both speaker-dependent and speaker-independent SAD. Although using the information of the whole face including the dynamics of eyes and cheeks seems intuitively useful, it remains to be demonstrated that whole face information is superior than mouth region features. A recent neural-based study by Shahid et al. (2019) found that using whole upper body features yields promising results —because the dynamics of the arms convey additional information while speaking. But it remains to be shown that upper body features are indeed a better choice than mouth or face regions.

Existing literature has shown various approaches in which SAD can be conducted using neural networks. It has also highlighted that neural networks can automate the identification of speech activity whilst outperforming other systems. CNNs in particular, have proven to have the ability to distinguish between speech and non-speech. Recently, the addition

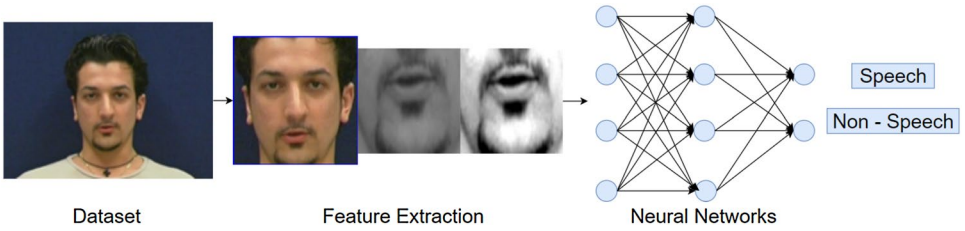

Fig. 1 Steps for studying visual recognition of speech activity

of RNNs has been shown to further increase the accuracy of the detection. However, the literature lacks studies in providing any comparison between these approaches (Fig. 1).

3 Visual speech detection

3.1 Experimental design

3.1.1 Dataset

Our experiments use the VidTIMIT dataset Sanderson and Lovell (2009). It consists of video and audio recordings of 43 individuals (19 females and 24 males) saying short phrases. Data was recorded over 3 sessions with a delay between each session to obtain different data (attire, hairstyle and beard etc.) for the same individual. Each session started with a head rotation sequence whilst images were captured. Following that, individuals were asked to recite some sentences whilst audio and visual data were recorded. For each individual and on average, 1346 images were captured during the head rotation sequence, and 1061 images were captured during the recitation. Thus, the total of head rotation images equates to 57881 and recitation images to 45661 for a total of 103,542 images. As a result, the dataset provides reasonable facial images for each category (speech and non-speech). Although our dataset contains audio-visual recordings, in this paper we only use visual information. Therefore, all our experiments are agnostic to acoustic noise.

In our experiments, the data is split in the ratio of 70:15:15 for training, validation and testing respectively. Some learning parameters across models include Adam as the optimizer (widely used), learning rate of 0.001, beta 1 and 2 at 0.9 and 0.999 respectively (as suggested in Kingma and Ba (2014)), ReLU activations, 3-fold cross-validation, 25 epochs, and Softmax classification.

3.1.2 Feature extraction

Extracting ROI images from the dataset involves labelling the data, categorising images into one of the two categories (speech/non-speech). The label attached to an image is based on two factors: the state of the image at the given time, and the state of the current image based on previous images.

As the dataset includes full-face images, images involving the mouth region were extracted via the Haar cascade algorithm (with OpenCV[1])—as illustrated in Fig. 2) and according to $MouthRegion = ey : ey + eh, ex : ex + (ew \times 2)$, where $ey = y + (eh \times 2)$, $ex = x + ew$, $eh = \frac{h}{3}$, $ew = \frac{w}{4}$, x and y represent the axes of the image, w and h represent the width and height of the image, and ex and ey are the starting points of the mouth-region as noted in Fig. 3.

The brightness and contrast of images are also altered so that images appear to be consistent. Due to lighting conditions, race (i.e. colour) or facial feature (e.g. beard), images can appear to be inconsistent which can produce unnecessary noise in the neural networks.

3.2 Detection of speech with still images

3.2.1 VGG-like model

VGG -visual geometry group Simonyan and Zisserman (2014) is one of the popular CNN models that has achieved top performances but also changed the way architectures are designed for the networks. Due to its simplicity, the architecture is adapted for the problem in question. However, as there are various versions of VGG (such as VGG-16, VGG-19) which carry a substantial number of layers and parameters, the architecture requires not only time but powerful computers that can train such models. This is demonstrated in Simonyan and Zisserman (2014) where the smallest VGG model has over 100 million parameters. As a result, the original architecture is adapted by a reduction in layers and sizes, see Fig. 4. It is denoted as

$$\mathbf{c}_1 = MAXPOOL(CONV(\mathbf{x})),$$
$$\mathbf{c}_2 = MAXPOOL(CONV(\mathbf{c}_1)),$$
$$\mathbf{c}_4 = MAXPOOL(CONV(CONV(\mathbf{c}_2))),$$
$$\mathbf{c}_6 = MAXPOOL(CONV(CONV(\mathbf{c}_4))),$$
$$\mathbf{y} = Softmax(\mathbf{W}''^{\mathsf{T}}(\mathbf{W}'^{\mathsf{T}}\mathbf{c}_6 + \mathbf{b}') + \mathbf{b}''),$$

[1] https://opencv.org

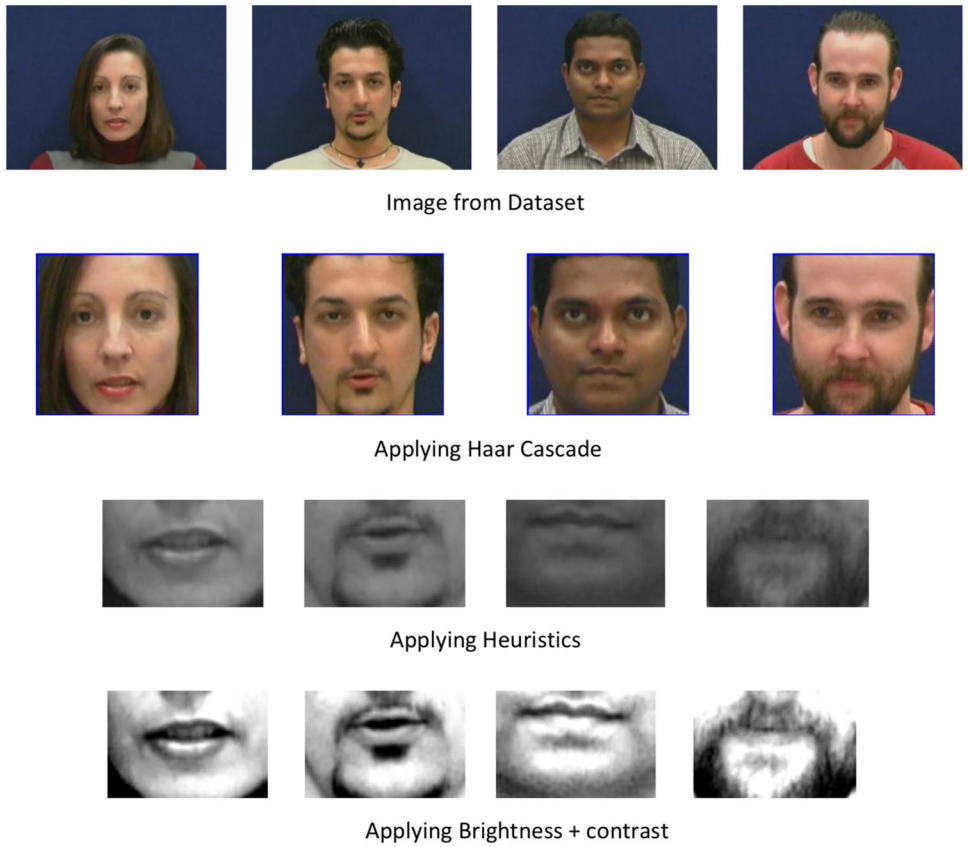

Fig. 2 Illustration of data pre-processing: raw data (top row), inputs to neural networks (bottom row)

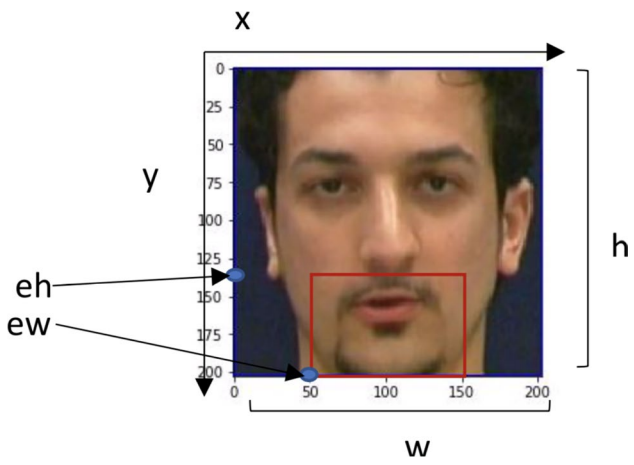

Fig. 3 Mouth-region extraction based on proposed procedure, see text for details

where $\mathbf{x}$ is the input image, $\mathbf{c}_i$ are the convolutional and pooling operations as in Goodfellow et al. (2016), $\mathbf{W}$ and $\mathbf{b}$ are the weights and biases in the fully connected (FC) layers, and *Softmax*() outputs a probability distribution of labels *notspeaking* and *speaking* according to $P(y_j) = \frac{e^{y_j}}{\sum_{k=1}^{K} e^{y_k}}$.

As the number of hidden units in the FC layers can affect the size of the network, different sizes are experimented (256, 512, 1024). Initial results showed that the model VGG-9 achieves a classification accuracy of 96% with 1024 hidden units in the FC layer. As a result, 1024 hidden neurons is the set parameter in the rest of our experiments.

Further experiments are conducted with the inclusion of batch normalization and dropout. Batch normalization is added before the activation function as recommended in Ioffe and Szegedy (2015). Dropout 0.2 and 0.5 is used in convolutional and FC layers, respectively. Such application is suggested by Srivastava et al. (2014) and applied by Le Cornu and Milner (2015); Ariav and Cohen (2019). Ariav and Cohen (2019) in particular, used a combination of both batch normalization and dropout. The stride in the first convolution in this model (VGG-9) is increased to 2. Enlarging the VGG-9 model in any shape or form albeit by adding batch normalization affects the resources required. As the network is reparameterized in each mini-batch with batch normalization, such a process can require additional computational resources considering the computation required for the convolution process. As a result, the stride of the first convolution is changed (stride = 2) to facilitate the memory required for the batch normalization operation.

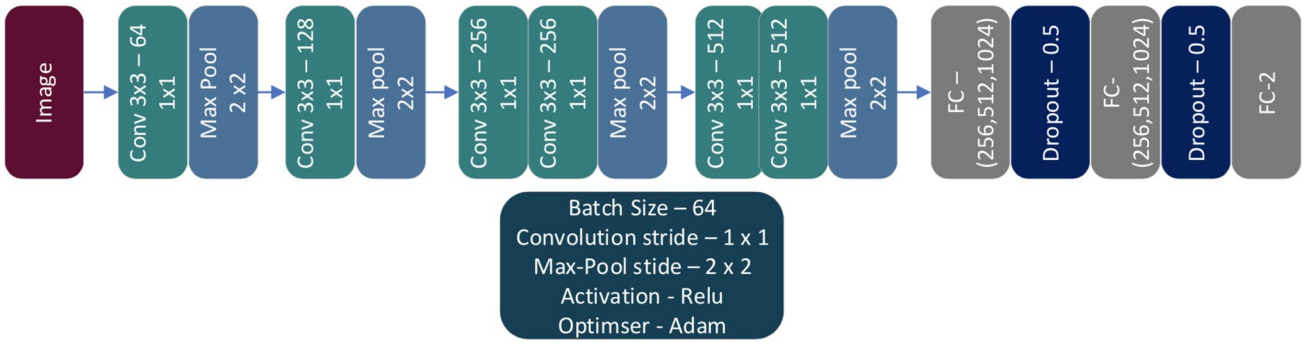

Fig. 4 VGG architecture (VGG-9)

3.2.2 Three-layered CNN

Based on Le Cornu and Milner (2015), Sharma et al. (2019) and Tao and Busso (2019), smaller networks—including 3 layers of convolution—are also experimented as a comparison. As commented in Sect. 2, smaller networks have proven to achieve reasonable accuracy. Adapting to smaller networks can also improve training and prediction times. In this case, the network still utilises 3 x 3 filters, and with an incremental number of filters (i.e 32, 64, 128). The same parameters and regularization techniques (batch normalization and dropout) are utilised as the VGG-9 architecture described previously.

3.3 Detection of speech with sequences of images

The previous subsection discussed the identification of speech via still images. That is, considering an image at a given time if the image belongs to the category of speech or non-speech. In this subsection, detection of speech is identified based on sequences of images. This section aims to find the effects of adding history/sequence of images in the detection of speech by comparing the performance of neural models with or without history. Two different but related architectures are studied: first, a sequence of images is classified based on previous images (referred to as *Encoder*); and second, classification of multiple images in a sequence (referred to as *Encoder-Decoder*).

3.3.1 Classification based on previous images: encoder

Sharma et al. (2019) and Wang and Wang (2019) used a combination of CNN and RNN to form a unified model. In this case, the CNN model may be an existing one (such as VGG or Xception) or may follow its architecture. This type of architecture is also referred to as CRNN (convolutional recurrent neural network.) The recurrent layers are used to encode the individual information of a sequence to support classification based on previous images. This architecture that takes recurrent layers into account is referred to as *Encoder*.

In this architecture, the CNN is derived from the architecture described in Sect. 3.2 ('3-Layered CNN'). Combining the 3-layered CNN with an RNN allows us to carry out a fair comparison, as well as to examine the effect of combining an RNN on top of a CNN, see Fig. 5. The CRNN is denoted as

$$\mathbf{c}_1 = MAXPOOL(CONV(\mathbf{x}_t)),$$
$$\mathbf{c}_2 = MAXPOOL(CONV(\mathbf{c}_1)),$$
$$\mathbf{c}_3 = MAXPOOL(CONV(\mathbf{c}_2)),$$
$$\mathbf{y} = Softmax(GRU(\mathbf{h}_{t-1}, \mathbf{c}_3)),$$

where $\mathbf{x}_t$ is the image at time t, $\mathbf{c}_i$ are the convolutional and pooling operations, *GRU* is a fast implementation of the RNN proposed by Cho et al. (2014)—from Chollet (2015)—to generate hidden states $\mathbf{h}_t$, and $\mathbf{y}$ are the output predictions.

Our encoder is implemented with a single GRU layer with 1024 hidden units. The GRUs of Cho et al. (2014) are employed—as opposed to the LSTMs of Hochreiter and Schmidhuber (1997)—as GRUs require less computation (due to fewer parameters) than LSTMs whilst achieving similar results Jozefowicz et al. (2015). Various authors, including Limet al. (2016), utilise 1024 units for recurrent layers to make the network small and compact. Our experiments are based on CuDNNGRUs as they provide up to 7.2x faster training than standard GRUs Braun (2018).

For CRNN models, the batch size is reduced to 32 as training with a batch size of 64 leads to slower training. Similarly, the sequence size for the CRNN models is up to sequence size of 10. Nonetheless, different sequence lengths are examined to find the optimal result and understand the difference in performance and accuracy. The CNN in this architecture is wrapped in a Time Distributed Layer, which introduces introducing a fifth dimension (time) and allows each image to be a timestamp in the sequence.

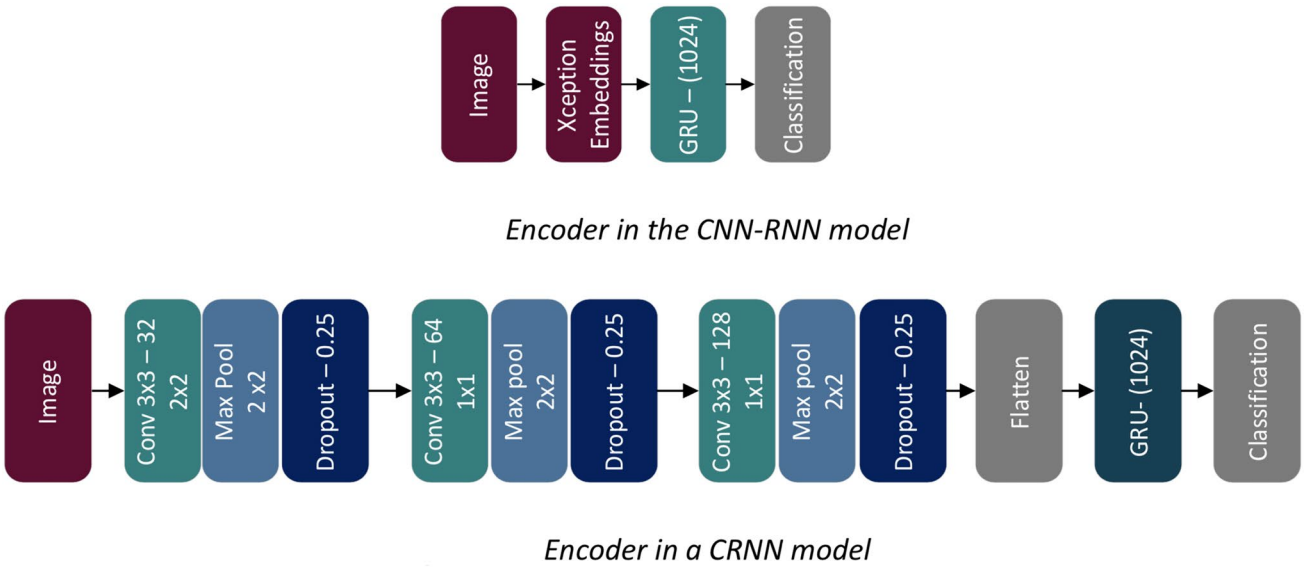

Fig. 5 The Encoder architectures

3.3.2 Classification of multiple images: encoder-decoder

Ariav and Cohen (2019), Soh (2016) and Wang et al. (2016) implemented an existing CNN model for image representation where models such as VGG are used to obtain embeddings from images. RNNs in the form of LSTMs are later attached for sequence learning. We refer to this architecture as CNN-RNN, which is usually associated with an encoder-decoder neural net. Soh (2016), Donahue et al. (2015), Venugopalan et al. (2015) and Vinyals et al. (2015) used a separate set of recurrent layers in which the first set of recurrent layer(s) act as an encoder, whilst the latter set of recurrent layer(s) pose as a decoder.

In our CNN-RNN models, the learned representations (image embeddings) are derived from Xception Chollet (2017). Xception is a more recent model based on Inception Szegedy et al. (2016) but with a smaller number of parameters and better classification accuracy than its predecessors. Opting for Xception would provide faster prediction times as it is one of the smallest models available. Furthermore, Xception has a higher top-1 and top-5 accuracy than other popular CNN models Filonenko et al. (2017).

However, Ariav and Cohen (2019) found that using existing CNNs may affect system performance as SAD is a binary classification, and large networks (such as VGG and Inception) are predominantly used for large datasets with a large number of classes. Thus, the Encoder and Encoder-Decoder models in our experiments are studied with both CNN-RNN and CRNN architectures to identify the impact of using embeddings from existing CNN architectures, see Fig. 6. The CRNN encoder-decoder is denoted as

$$
\begin{aligned}
\mathbf{c}_1 &= MAXPOOL(CONV(\mathbf{x}_t)),\\
\mathbf{c}_2 &= MAXPOOL(CONV(\mathbf{c}_1)),\\
\mathbf{c}_3 &= MAXPOOL(CONV(\mathbf{c}_2)),\\
\mathbf{h}_t^{enc} &= BatchNorm(GRU(\mathbf{h}_{t-1}^{enc}, \mathbf{c}_3),\\
\mathbf{h}_t^{dec} &= BatchNorm(GRU(\mathbf{h}_{t-1}^{dec}, \mathbf{h}_t^{enc}),\\
\mathbf{y} &= Softmax(\mathbf{h}_t^{dec}),
\end{aligned}
$$

where $\mathbf{x}_t$ is the input image at time t, $\mathbf{c}_i$ are the convolutional and pooling operations, *GRU* is a fast implementation of the RNN proposed by Cho et al. (2014)—from Chollet (2015)—to generate hidden states $\mathbf{h}_t$, *BatchNorm* normalises the learned weights as in Ioffe and Szegedy (2015), and $\mathbf{y}$ are the output predictions.

Like the encoder, the encoder-decoder uses GRUs with 1024 hidden units to keep the models comparable. In terms of the number of GRU cells, this is dependent on the length of the sequence. Li et al. (2019) found that batch normalization is useful for Encoder-Decoder architectures. As a result batch normalization is applied to all Encoder-Decoder models in our experiments.

The batch size for CNN-RNNs is kept the same as for CNNs at 64, as increasing the batch size any further can cause memory-related issues, especially for larger sequences. To keep the models comparable, sequence length for CNN-RNNs is up to 10 with various sizes experimented (PC used in all experiments: CPU = Intel Skylate i7 - 6700HQ, GPU = NVidia GTX 960M, RAM = 16GB).

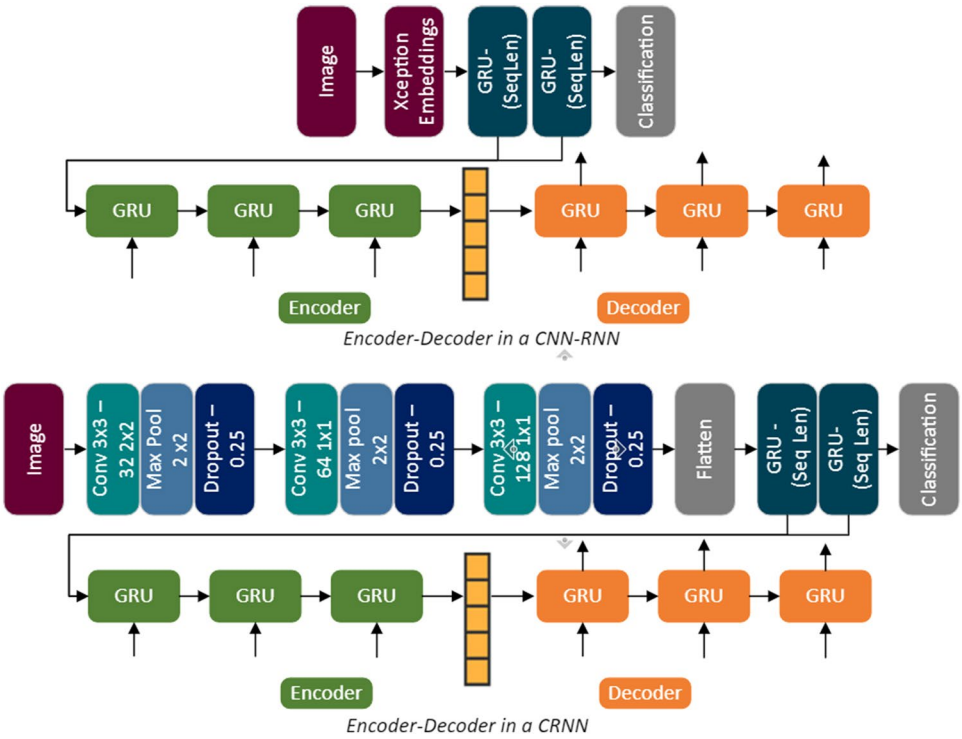

Fig. 6 The Encoder-Decoder architecture

Table 1 Comparison of results of the two CNN architectures

Properties/architecture	VGG-9 (adjusted, see Sect. 3.2.1)	3-Layered CNN (see Sect. 3.2.2)
Number of conv layers	6 Layered	3-layered
Filter size	64,128,256,256,512,512	32,64,128
Normalization/	Batch norm	
Regularization	Dropout(0.2)	Dropout(0.25,0.5)
Classification accuracy	0.9708	0.9721
Train time (on avg. per epoch)	163.16 s	155.92 s
Test time (on avg. per image)	9.69 ms	5.865 ms

4 Results

4.1 Results of still images

In Table 1, the architecture on the left shows the adjusted VGG-9 architecture described in Sect. 3.2. It achieved an accuracy of 97.08%. The average training time was 163.16 seconds (per epoch) with an average prediction time of 9.69 milliseconds. With the addition of batch normalization and dropout, there is an increase in accuracy of 1% from the initial model mentioned previously.

With the smaller network (3-layered CNN), batch normalization seemed useful in increasing the speed of learning as suggested by Ioffe and Szegedy (2015), but did not aid in the performance of the network and thus batch normalization was discarded. Similarly, our experimental results show that in a smaller network, having multiple FC layers does not improve the accuracy, but with a single FC layer the network performs better. Ultimately, the smaller network (3-layered CNN) achieved the highest accuracy of 97.21% with a loss of 0.075. This smaller architecture outperforms its counterpart (VGG-9) in all aspects including classification accuracy and training & prediction times.

4.2 Results of sequences of images

Table 2 shows classification accuracies of Encoders and Encoder-Decoders with varying sequence lengths. As noted, all four architectures achieve reasonable results and there is a marginal difference between the accuracy of the models. However, for sequence-based classification, the results show that the sequence size needs to be at least 3

Table 2 Classification results of four neural networks with varying sequence lengths

	Encoder	Encoder-decoder
CNN-RNN		
Seq len 1	0.9465	0.9391
Seq len 3	0.9697	0.9557
Seq len 5	0.9810	0.9642
Seq len 10	0.9857	0.9694
CRNN		
Seq len 1	0.9639	0.9611
Seq len 3	0.9880	0.9846
Seq len 5	**0.9914**	0.9915
Seq len 10	0.9892	**0.9961**

or more to outperform the CNN classification accuracy of 97.21% (reported above). Furthermore, with sequences of images ≥ 5, all architectures outperform the CNN models.

With regard to image classification based on previous images, the CRNN architecture achieved the highest accuracy of 99.14%. Its counterpart for sequence classification—CRNN with Encoder-Decoder—achieved the highest accuracy of 99.61%. These results show that in this problem domain, using large CNN architectures for feature learning can decrease accuracy over using smaller (and domain-specific) architectures for sequence classification.

It can also be noted that the CRNN-based encoder-decoder architecture does provide the highest network accuracy but requires an increase in history (sequence size). At lower sequence sizes, the Encoder models achieve greater accuracy compared to the Encoder-Decoder models.

Setting the sequence length as a hyperparameter highlighted that increasing the sequence length has a positive effect on the accuracy of the network (as the network has more data in history for classification). However, in the case of CRNNs at sequence length 10, there is no increase in accuracy from the previous size. Zhang et al. (2017) found that after a certain size increase in sequence size can make it more difficult for the network to predict the right output. Batch normalization was found useful for the Encoder-Decoder models as it reduced overfitting and increased accuracy. However, for the Encoder models, batch normalization did not prove to be as beneficial.

Inspired by Sharma et al. (2019), further experiments were carried out by increasing the convolutional stride to make the network smaller, thus resulting in faster predictions. Sehgal and Kehtarnavaz (2018) compared different strides and found that increasing stride by more than 2 can cause a network to be unstable and a noticeable reduction in classification accuracy. As a result, the stride of the CRNN Encoder architecture was increased to 2. The result of this experiment showed a reduction in training times by half but caused a 2% reduction in classification accuracy.

Table 3 Comparison of the four architectures achieving the highest accuracy (best in bold)

Model	VGG-9 (Adjusted)	3-layered CNN	CRNN Encoder (Seq len:5)	bf CRNN Encoder-decoder (Seq len:10)
Number of Parameters	10,286,082	18,970,114	**17,402,370**	**23,708,162**
Classification Accuracy	0.9708	0.9721	**0.9914**	**0.9961**
Avg. training Time (per epoch)	163.16 s	155.92 s	**325.72 s**	**761 s**
Avg. prediction Time (per image)	9.69 ms	5.86 ms	**3.7 ms**	**5.19 ms**

5 Discussion

Table 3 compares the architectures that provided the highest accuracy for the task of SAD. As noted, the introduction of history increases classification accuracy, as CRNN-based Encoder and Encoder-Decoder architectures achieve 99% accuracy. Furthermore, with RNNs ability of memory, CNN models can be designed smaller which allows for even faster prediction times.

From our results in Table 3, it can be noted that classification based on previous images using the CRNN-based Encoder architecture provides an accuracy of 99.14% with 3.7ms for prediction. On the other hand, the classification of sequences of images using the CRNN-Based Encoder-Decoder provides the highest accuracy of 99.61% with 5.19ms for prediction times. Both of these architectures outperform CNN architectures but require a history of images. Although the Encoder-Decoder model provides the highest accuracy of 99.61% it, takes almost twice as long to train and predict whereas the CRNN Encoder achieves 99.14% but offers the fastest training and prediction times. The latter is especially important for applications that require near-real-time performance. This suggests that it is not worth the effort in using sequences of length 10 and that sequences of length 5 should be preferred in neural architectures applied to the task of SAD. In our experiments, we found marginal differences in the classification performance for females and males.

Results from Tables 2 and 3 highlight that using smaller networks over embeddings from existing CNN architectures,

provide better accuracy and offer faster predictions. The CNN-RNN, in this case, took 30ms for prediction as opposed to the variants of CRNN that took between 3 and 5ms. The training times between encoder and encoder-decoder are significant due to the architectures and sequence sizes. The Encoder is a simpler network (with a single GRU layer) compared to the encoder-decoder which is a more complex network (consisting of encoding and decoding layers). In summary, our comparison of visual-based recognisers for the task of speech activity detection provides evidence to suggest that a CRNN-based Encoder architecture is the best compromise between classification accuracy and training/test times.

6 Conclusion and future work

This article studies speech activity detection (SAD) using three types of neural architectures: classification with still images (CNN), classification based on previous images (encoder), and classification of sequences of images (encoder-decoder). CNNs are considered to obtain baseline performances with two CNN variant architectures. For comparison, the use of sequences of images is also experimented. Encoder and encoder-decoder architectures were combined and compared with CNN-RNN and CRNN networks.

Regarding still images, our results show that the smaller CNN provides a higher accuracy at 97.21% whilst offering faster training and prediction times than a VGG-9 model. Regarding sequences of images, results showed that both RNN architectures can outperform the CNN baselines as they achieve 99% compared to 97% for CNNs. Furthermore, our results show that RNN-based architectures can be as fast predictors (or even faster) than CNNs.

Depending on the requirements and computational resources available, both of these architectures can be considered for real-time application. Although a simpler CRNN Encoder may not achieve the best accuracy than a complex network (such as an encoder-decoder), the network is still a justified choice. This is due to its ability to be a near top classifier and to provide faster prediction and training times than encoder-decoders.

Future work involves deployments of top speech activity detectors in challenging scenarios such as car infotainment systems or robots interacting with humans in noisy environments. Experiments using several datasets instead of one or two is also research that remains to be explored, which would reveal further information regarding the performance and ranking of neural architectures across datasets. Whilst the work above focuses on offline training, future work could study online training for improved system performance—where fast training can be relevant. Last but not least, detection may utilise video and audio-based recognition from different sensors and tracking different/multiple parts of the body (mouth, face, whole upper body) for more robust operation in changing environments (e.g. with good/poor illuminating conditions).

References

Ariav, I., & Cohen, I. (2019). An end-to-end multimodal voice activity detection using wavenet encoder and residual networks. *IEEE Journal of Selected Topics in Signal Processing, 13*(2), 265–274.

Braun, S. (2018). LSTM benchmarks for deep learning frameworks. arXiv preprint arXiv:180601818

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Moschitti, A., Pang, B., Daelemans, W. (Eds.) Conference on empirical methods in natural language processing (EMNLP), ACL

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. IEEE conference on computer vision and pattern recognition

Chollet. F., et al. (2015) Recurrent neural networks (rnn) with keras. https://www.tensorflow.org/guide/keras/rnn

Cooke, M., Barker, J., Cunningham, S., & Shao, X. (2006). An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America, 120*(5), 2421–2424.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In: IEEE conference on computer vision and pattern recognition

Filonenko, A., Kurnianggoro, L., Jo, K.H. (2017). Comparative study of modern convolutional neural networks for smoke detection on image data. In: International Conference on Human System Interactions (HSI), IEEE

Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press, http://www.deeplearningbook.org

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:150203167

Joosten, B., Postma, E. O., & Krahmer, E. (2015). Voice activity detection based on facial movement. *Journal of Multimodal User Interfaces, 9*(3), 183–193.

Jozefowicz, R., Zaremba, W., Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In: International conference on machine learning

Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:14126980

Le Cornu, T., Milner, B. (2015). Voicing classification of visual speech using convolutional neural networks. In: FAAVSP-the 1st joint conference on facial analysis, animation and auditory-visual speech processing

Li, A., Zheng, C., Li, X. (2019). Convolutional recurrent neural network based progressive learning for monaural speech enhancement. arXiv:190810768

Lim, W., Jang, D., Lee, T. (2016). Speech emotion recognition using convolutional and recurrent neural networks. In: Asia-Pacific Signal and Inf. Proceedings of the association annual summit and conference (APSIPA).

Sanderson, C., Lovell, B. C. (2009). Multi-region probabilistic histograms for robust and scalable identity inference. In: International conference on biometrics, Springer.

Sehgal, A., & Kehtarnavaz, N. (2018). A convolutional neural network smartphone app for real-time voice activity detection. *IEEE Access, 6,* 9017–9026.

Shahid, M., Beyan, C., Murino, V. (2019). Voice activity detection by upper body motion analysis and unsupervised domain adaptation. In: IEEE/CVF international conference on computer vision workshops.

Sharma, T., Aralikatti, R., Margam, D. K., Thanda, A., Roy, S., Kandala, P.A., Venkatesan, S. M. (2019). Real time online visual end point detection using unidirectional LSTM. Interspeech.

Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Soh, M. (2016). *Learning CNN-LSTM architectures for image caption generation*. Technical Report: Stanford University.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15*(1), 1929–1958.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In: IEEE conference on computer vision and pattern recognition.

Tao, F., & Busso, C. (2019). End-to-end audiovisual speech activity detection with bimodal recurrent neural models. *Speech Communication*. https://doi.org/10.1016/j.specom.2019.07.003.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K. (2015). Sequence to sequence-video to text. In: IEEE international conference on computer vision.

Vinyals, O., Toshev, A., Bengio, S., Erhan, D. (2015). Show and tell: A neural image caption generator. In: IEEE conference on computer vision and pattern recognition.

Wang, B., Wang, X. (2019). Are you speaking: Real-time speech activity detection via landmark pooling network. In: 2019 14th IEEE international conference on automatic face & gesture recognition (FG 2019).

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W. (2016). CNN-RNN: A unified framework for multi-label image classification. In IEEE conference on computer vision and pattern recognition.

Zhang, Y., Sun, X., Ma, S., Yang, Y., Ren, X. (2017). Does higher order LSTM have better accuracy for segmenting and labeling sequence data? arXiv:171108231