Audio Visual Speech Recognition Using Deep Recurrent Neural Networks

Abhinav Thanda(✉) and Shankar M. Venkatesan

Samsung R&D Institute India, Bangalore, Bangalore, India
{abhinav.t89,s.venkatesan}@samsung.com

Abstract. In this work, we propose a training algorithm for an audio-visual automatic speech recognition (AV-ASR) system using deep recurrent neural network (RNN). First, we train a deep RNN acoustic model with a Connectionist Temporal Classification (CTC) objective function. The frame labels obtained from the acoustic model are then used to perform a non-linear dimensionality reduction of the visual features using a deep bottleneck network. Audio and visual features are fused and used to train a fusion RNN. The use of bottleneck features for visual modality helps the model to converge properly during training. Our system is evaluated on GRID corpus. Our results show that presence of visual modality gives significant improvement in character error rate (CER) at various levels of noise even when the model is trained without noisy data. We also provide a comparison of two fusion methods: feature fusion and decision fusion.

Keywords: Audio-visual speech recognition · Connectionist Temporal Classification · Recurrent neural network

1 Introduction

Audio-visual automatic speech recognition (AV-ASR) is a case of multi-modal analysis in which two modalities (audio and visual) complement each other to recognize speech. Incorporating visual features, such as speaker's lip movements and facial expressions, into automatic speech recognition (ASR) systems has been shown to improve their performances especially under noisy conditions. To this end several methods have been proposed which traditionally included variants of GMM/HMM models [3,5]. More recently AV-ASR methods based on deep neural networks (DNN) [14,21,23] have been proposed.

End-to-end speech recognition methods based on RNNs trained with CTC objective function [10,11,19] have come to the fore recently and have been shown to give performances comparable to that of DNN/HMM. The RNN trained with CTC directly learns a mapping between audio feature frames and character/phoneme sequences. This method eliminates the need for an intermediate step of training GMM/HMM model, thereby simplifying the training procedure. To our knowledge, so far AV-ASR systems based on RNN trained with CTC have not been explored.

F. Schwenker and S. Scherer (Eds.): MPRSS 2016, LNAI 10183, pp. 98–109, 2017.
DOI: 10.1007/978-3-319-59259-6_9

In this work, we design and evaluate an audio-visual ASR (AV-ASR) system using deep recurrent neural network (RNN) and CTC objective function. The design of an AV-ASR system includes the tasks of visual feature engineering, and audio-visual information fusion. Figure 1 shows the AV-ASR pipeline at test time. This work mainly deals with the visual feature extraction and processing steps and training protocol for the fusion model. Proper visual features are important especially in the case of RNNs as RNNs are difficult to train. Bottleneck features used in tandem with audio features are known to improve ASR performance [7,12,28]. We employ a similar idea in order to improve the discriminatory power of video features. We show that this helps the RNN to converge properly when compared with raw DCT features. Finally, we compare the performances of feature fusion and decision fusion methods.

The paper is organized as follows: Sect. 2 presents the prior work on AV-ASR. Bi-directional RNN and its training using CTC objective function are discussed in Sect. 3. Section 4 describes the feature extraction steps for audio and visual modalities. In Sect. 5 different fusion models are explained. Section 6 explains the training protocols and experimental results. Finally, we summarize our work in Sect. 7.

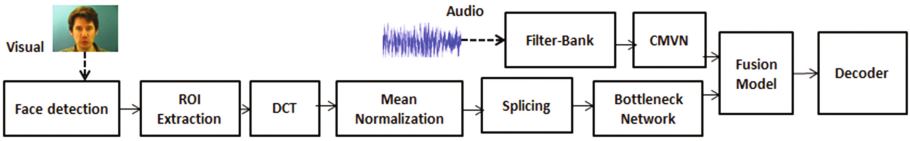

Fig. 1. Pipeline of AV-ASR system at test time. Fusion

2 Related Work

The differences between various AV-ASR systems lie chiefly in the methods employed for visual feature extraction and audio-visual information fusion. Visual feature extraction methods can be of 3 types [24]: 1. Appearance based features where each pixel in the mouth region of the speaker (ROI) is considered to be informative. Usually a transformation such as DCT or PCA is applied to the ROI to reduce the dimensions. Additional feature processing such as mean normalization, intra-frame and inter-frame LDA may be applied [15,24]. 2. Shape based features utilize the geometric features such as height, width and area of the lip region or build a statistical model of the lip contours whose parameters are used as features. 3. Combination of appearance and shape based features.

Fusion methods can be broadly divided into two types [16,24]: 1. Feature fusion 2. Decision fusion. Feature fusion models perform a low level integration of audio and visual features and this involves a single model which is trained on the fused features. Feature fusion may include a simple concatenation of features or feature weighting and is usually followed by a dimensionality reduction

transformation like LDA. On the other hand, Decision fusion is applied in cases where the output classes for the two modalities are same. Various decision fusion methods based on variants of HMMs have been proposed [3,5]. In Multistream HMM the emission probability of a state of audio-visual system is obtained by a linear combination of log-likelihoods of individual streams for that state. The parameters of HMMs for individual streams can be estimated separately or jointly. While multistream HMM assumes state level synchrony between the two streams, some methods [2,3] such as coupled HMM [3] allow for asynchrony between two streams. For a detailed survey on HMM based AV-ASR systems we refer the readers to [16,24]

Application of deep learning to multi-modal analyses was presented in [22] which describes multi-modal, cross-modal and shared representation learning and their applications to AV-ASR. In [14], Deep Belief Networks (DBN) are explored. In [21] the authors train separate networks for audio and visual inputs and fuse the final layers of two networks, and then build a third DNN with the fused features. In addition, [21] presents a new DNN architecture with a bilinear soft-max layer which further improves the performance. In [23] a deep de-noising auto-encoder is used to learn noise robust speech features. The auto-encoder is trained with MFCC features of noisy speech as input and reconstructs clean features. The outputs of final layer of the auto-encoder are used as audio features. A CNN is trained with images from the mouth region as input and phoneme labels as output. The final layers of the two networks are then combined to train a multi-stream HMM.

3 Sequence Labeling Using RNN

The following notations are adopted in this paper. For an utterance u of length T_u, $\mathbf{O}_a^u = (\overline{O}_{a,1}^u, \overline{O}_{a,2}^u, ..., \overline{O}_{a,T_u}^u)$ and $\mathbf{O}_v^u = (\overline{O}_{v,1}^u, \overline{O}_{v,2}^u, ..., \overline{O}_{v,T_u}^u)$ denote the observation sequences of audio and visual frames where $\overline{O}_{a,t} \in \mathbb{R}^{d_a}$ and $\overline{O}_{v,t} \in \mathbb{R}^{d_v}$. We assume equal frame rates for audio and visual inputs which is ensured in experiments by means of interpolation. $\mathbf{O}_{av}^u = (\overline{O}_{av,1}^u, \overline{O}_{av,2}^u, ..., \overline{O}_{av,T_u}^u)$ where $\overline{O}_{av,t}^u = [\overline{O}_{a,t}^u, \overline{O}_{v,t}^u] \in \mathbb{R}^{d_{av}}$ where $d_{av} = d_a + d_v$ denotes the concatenated features at time t for utterance u. The corresponding label sequence is given by $l = (l_1, l_2, ..., l_{S_u})$ where $S_u \leq T_u$ and $l_i \in L$ where L is the set of English letters and an additional element representing a space. For ease of representation, we drop the utterance index u. All the models described in this paper are character based.

3.1 Bi-directional RNN

RNNs are a class of neural networks used to map sequences to sequences. This is possible because of the feedback connections between hidden nodes. In a bi-directional RNN, the hidden layer has two components each corresponding to forward (past) and backward (future) connections. For a given input sequence

$\mathbf{O} = (\overline{O}_1, \overline{O}_2, ..., \overline{O}_T)$, the output of the network is calculated as follows: forward pass through forward component of the hidden layer at a given instant t is given by

$$\overline{h}_t^f = g(\mathbf{W}_{ho}^f \overline{O}_t + \mathbf{W}_{hh}^f \overline{h}_{t-1}^f + \overline{b}_h^f) \tag{1}$$

where $\mathbf{W}_{ho}^f$ is the input-to-hidden weights for forward component, $\mathbf{W}_{hh}^f$ corresponds to hidden-to-hidden weights between forward components, and $\overline{b}_h^f$ is the forward component bias. g is a non-linearity depending on the choice of the hidden layer unit. Similarly, forward pass through the backward component of the hidden layer is given by

$$\overline{h}_t^b = g(\mathbf{W}_{ho}^b \overline{O}_t + \mathbf{W}_{hh}^b \overline{h}_{t-1}^b + \overline{b}_h^b) \tag{2}$$

where $\mathbf{W}_{ho}^b$, $\mathbf{W}_{hh}^b$, $\overline{b}_h^b$ are the corresponding parameters for the backward component. The input to next layer is the concatenated vector $[\mathbf{h}_t^f, \mathbf{h}_t^b]$. In a deep RNN multiple such bidirectional hidden layers are stacked.

RNNs are trained using Back-Propagation Through Time (BPTT) algorithm. The training algorithm suffers from vanishing gradients problem which is overcome by using a special unit in hidden layer called the Long Short Term Memory (LSTM) [8,13].

3.2 Connectionist Temporal Classification

DNNs used in ASR systems are frame-level classifiers i.e., each frame of the input sequence requires a class label in order for the DNN to be trained. The frame-level labels are usually HMM states, obtained by first training a GMM/HMM model and then by forced alignment of input sequences to the HMM states. CTC objective function [9,10] obviates the need for such alignments as it enables the network to learn over all possible alignments.

Let the input sequence be $\mathbf{O} = (\overline{O}_1, \overline{O}_2, ..., \overline{O}_T)$ and a corresponding label sequence $\mathbf{l} = (l_1, l_2, ..., l_S)$ where $S \leq T$. The RNN employs a soft-max output layer containing one node for each element in L' where $L' = L \cup \{\phi\}$. The number of output units is $|L'| = |L| + 1$. The additional symbol ϕ represents a blank label meaning that the network has not produced an output for that input frame. The additional blank label at the output allows us to define an alignment π of length T containing elements of L'. For example, $(A\phi\phi M\phi), (\phi A\phi\phi M)$ are both alignments of length 5 for the label sequence AM. Accordingly, a many to one map $B : L'^T \longmapsto L^{\leq T}$ can be defined which generates the label sequence from an alignment.

Assuming that the posterior probabilities obtained at soft-max layer, at each instant are independent we get

$$P(\pi|\mathbf{O}) = \prod_{t=1}^{T} P(k_t|\overline{O}_t) \tag{3}$$

where $k \in L'$ and

$$P(k_t|\overline{O}_t) = \frac{\exp(y_t^k)}{\Sigma_{k'} \exp(y_t^{k'})} \tag{4}$$

where y_t^k is the input to node k of the soft-max layer at time t

The likelihood of the label sequence given an observation sequence can be calculated by summing (3) over all possible alignments.

$$P(\mathbf{l}|\mathbf{O}) = \sum_{\pi \in B^{-1}(\mathbf{l})} P(\pi|\mathbf{O}) \tag{5}$$

The goal is to maximize the log-likelihood $\log P(\mathbf{l}|\mathbf{O})$ estimation of a label sequence given an observation sequence. Equation 5 is computationally intractable since the number of alignments increases exponentially with the number of labels. For efficient computation of (5), forward-backward algorithm is used.

4 Feature Extraction

4.1 Audio Features

The sampling rate of audio data is converted to 16 kHz. For each frame of speech signal of 25 ms duration, filter-bank features of 40 dimensions are extracted. The filter-bank features are mean normalized and Δ and $\Delta\Delta$ features are appended. The final 120 dimensional features are used as audio features.

4.2 Visual Features

The video frame rate is increased to match the rate of audio frames through interpolation. For AV-ASR, the ROI for visual features is the region surrounding the speaker's mouth. Each frame is converted to gray scale and face detection is performed using Viola-Jones algorithm. The 64×64 lip region is extracted by detecting 68 landmark points [17] on the speakers face, and cropping the ROI surrounding speakers mouth and chin. 100 dimensional DCT features are extracted from the ROI.

After several experiments of training with DCT features, we found that RNN training either exploded or converged poorly. In order improve the discriminatory power of the visual features, we perform non-linear dimensionality reduction of the features using a deep bottleneck network. Bottleneck features are obtained by training a neural network in which one of the hidden layers has relatively small dimension. The DNN is trained using cross-entropy cost function with character labels as output. The frame-level character labels required for training the DNN are obtained by first training an acoustic model (RNN_a) and then obtaining the outputs from the final soft-max layer of RNN_a.

The DNN configuration is given by $dim - 1024 - 1024 - 40 - 1024 - opdim$ where $dim = 1100$ and is obtained by splicing each 100 dimensional video frame

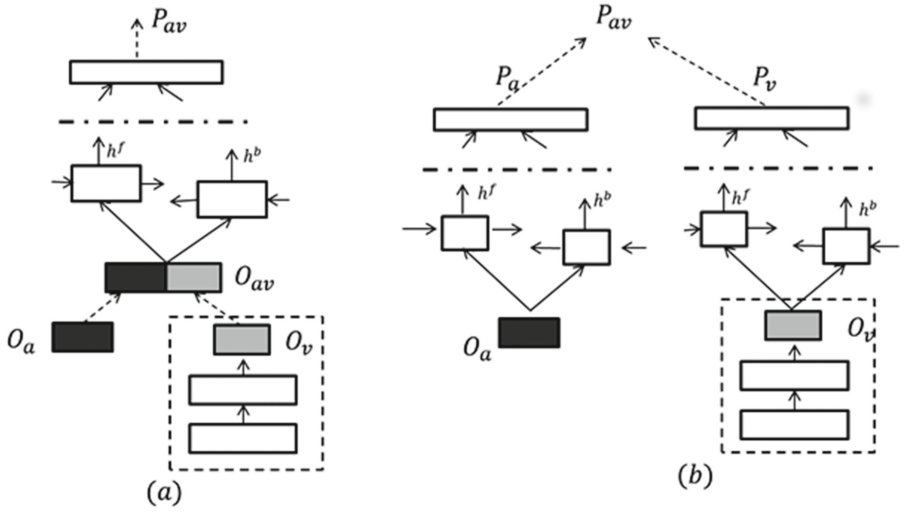

Fig. 2. Fusion models (a) Feature fusion (b) Decision fusion. The bottleneck network for visual feature extraction is enclosed in the dotted box.

with a context of 10 frames - 5 on each side. $opdim = |L'|$. After training, the last 2 layers are discarded and 40-dimensional outputs are used as visual features. The final dimension of visual feature vector is 120 including the Δ and $\Delta\Delta$ features.

5 Fusion Models

In this work, the fusion models are character based RNNs trained using CTC objective function i.e. L' is the set of English alphabet including a blank label. The two fusion models are shown in Fig. 2.

5.1 Feature Fusion

In feature fusion technique, a single RNN_{av} is trained by concatenating the audio and visual features using the CTC objective function. In the test phase, at each instant the concatenated features are forward propagated through the network. In the CTC decoding step, the posterior probabilities obtained at the soft-max layer are converted to pseudo log-likelihoods [26] as

$$\log P_{av}(\overline{O}_{av,t}|k) = \log P_{av}(k|\overline{O}_{av,t}) - \log P(k) \tag{6}$$

where $k \in L'$ and $P(k)$ is the prior probability of class k obtained from the training data [19].

5.2 Decision Fusion

In decision fusion technique the audio and visual modalities are modeled by separate networks, RNN_a and RNN_v respectively. RNN_v is a lip-reading system. The networks are trained separately. In the test phase, for a given utterance the frame level, the pseudo log-likelihoods of RNN_a and RNN_v are combined as

$$\log P_{av}(\overline{O}_{a,t}, \overline{O}_{v,t}|k) = \gamma \log P_a(k|\overline{O}_{a,t}) + (1-\gamma)\log P_v(k|\overline{O}_{v,t}) - \log P(k) \quad (7)$$

where $0 \leq \gamma \leq 1$ is a parameter dependent on the noise level and the reliability of each modality [5]. For example, at higher levels of noise in audio input, a low value of γ is preferred. In this work, we adapt the parameter γ for each utterance based on KL-divergence measure between the posterior probability distributions of RNN_a and RNN_v. The divergence between the posterior probability distributions is expected to vary as the noise in the audio modality increases. The KL-divergence is scaled to a value in $[0, 1]$ using logistic sigmoid. The parameter b was determined empirically from validation dataset.

$$D_{KL}(P_v||P_a) = \sum_i P_v log P_a \quad (8)$$

where we consider the posteriors of RNN_v as the true distribution based on the assumption that video input is always free from noise.

$$\gamma = \frac{1}{1 + exp(-D_{KL} + b)} \quad (9)$$

6 Experiments

The system was trained and tested on GRID audio-visual corpus [4]. GRID corpus is a collection of audio and video recordings of 34 speakers (18 male, 16 female) each uttering a 1000 sentences. Each utterance has a fixed length of approximately 3 s. The total number of words in the vocabulary is 51. The syntactic structures of all sentences are similar as shown below.

$< command > \ < color > \ < preposition > \ < letter > \ < digit > \ < adverb >$
Ex. PLACE RED AT M ZERO PLEASE

6.1 Training

In the corpus obtained, the video recordings for speaker 21 were not available. In addition, 308 utterances by various speakers could not be processed due to various errors. The dataset in effect consisted of 32692 utterances 90% of the which (containing 29423 utterances) was used for training and cross validation while the remaining (10%) data was used as test set. Both training and test data contain utterances from all of the speakers. Models were trained and tested using Kaldi speech recognition tool kit [25], Kaldi+PDNN [18] and EESEN framework [19].

***RNN*$_a$ Acoustic Model.** RNN_a contains 2 bi-directional LSTM hidden layers. Input to the network is 120-dimensional vector containing filter-bank coefficients along with Δ and $\Delta\Delta$ features. The model parameters are randomly initialized within the range $[-0.1, 0.1]$. The initial learning rate is set to 0.00004. Learning rate adaption is performed as follows: when the improvement in accuracy on the cross-validation set between two successive epochs falls below 0.5%, the learning rate is halved. The halving continues for each subsequent epoch until the training stops when the increase in frame level accuracy is less than 0.1%.

Deep Bottleneck Network. The training protocol similar to [26] was followed to train the bottleneck network. Input video features are mean normalized and spliced. Cross-entropy loss function is minimized using mini-batch Stochastic Gradient Descent (SGD). The frames are shuffled randomly before each epoch. Batch size is set to 256 and initial learning rate is set to 0.008. Learning rate adaptation similar to acoustic model is employed.

***RNN*$_v$-Lip Reader.** RNN_v is trained with bottleneck network features as input. The network architecture and training procedure is same as RNN_a. Figure 3 depicts the learning curves when trained with bottleneck features and DCT features. The figure shows that bottleneck features are helpful in proper convergence of the model.

***RNN*$_{av}$.** The feature fusion model RNN_{av} consists of 3 bi-directional LSTM hidden layers. The input dimension is 240, corresponding to filter-bank coefficients of audio modality, bottleneck features of visual modality and their respective Δ features. The initialization and learning rate adaption are similar to acoustic model training. However, the learning rate adaptation is employed only after a minimum number of (in this case 20) epochs are completed.

During each utterance in an epoch we first present the fused audio-visual fused input sequence followed by the input sequence with audio input set to very low values. This prevents the RNN_{av} from over-fitting to audio only inputs. Thus the effective number of sequences presented to the network in a given epoch is twice the total number of training utterances (AV and V features). After the training with AV and V features we train the network once again with two epochs of audio only utterances obtained by turning off the visual modality.

6.2 Results

The audio-visual model is tested with three levels of babble noise 0 dB SNR, 10 dB SNR and clean audio. Noise was added to test data artificially by mixing babble noise with clean audio .wav files. In order to show the importance of visual modality under noisy environment, the model is tested with either audio or video inputs turned off. A token WFST [19] is used to map the paths to their corresponding label sequences. The token WFST obtains this mapping by removing all the blanks and repeated labels. Character Error Rate (CER) is obtained from the decoded and expected label sequences by calculating the edit distance between them. The CER results are shown in Table 1.

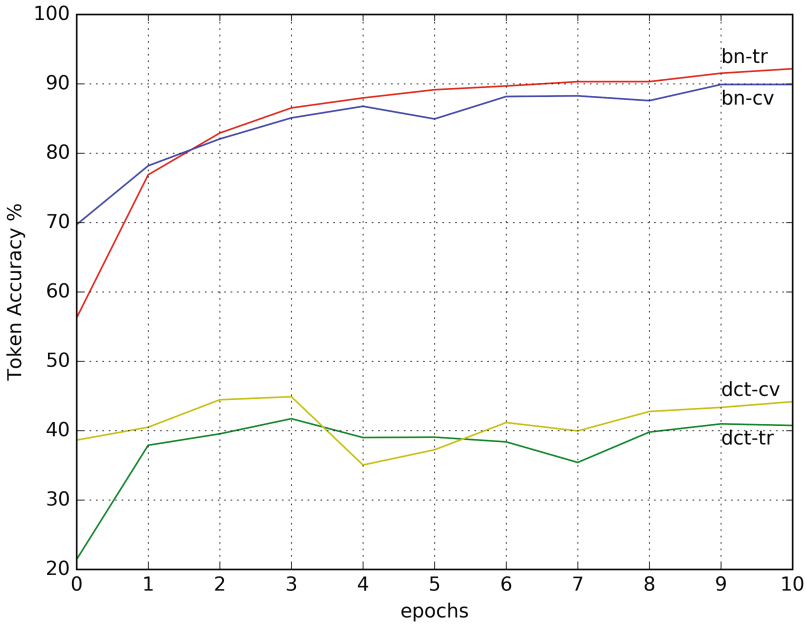

Fig. 3. Learning curves for bottleneck (bn) features and DCT features for training (tr) and validation (cv) data sets.

We observe that with clean audio input, audio only RNN_a performs significantly better (CER 2.45%) compared to audio-visual RNN_{av} (CER 5.74%). However as audio becomes noisy, the performance of RNN_a deteriorates significantly whereas the performance of RNN_{av} remains relatively stable. Under noisy conditions the feature fusion model behaves as if it is not receiving any input from the audio modality.

Table 1 also gives a comparison between feature fusion model and decision fusion model. We find that feature fusion model performs better than decision fusion model in all cases except under clean audio conditions. The poor CER of RNN_a, RNN_v model indicates that the frame level predictions between RNN_a and RNN_v are not synchronous. However, both the fusion models provide significant gains under noisy audio inputs. While there is large difference between RNN_a and other models with clean inputs, we believe this difference is due to the nature of dataset and will reduce with larger datasets.

Comparison with Lip-Reading Systems. While a number of AV-ASR models exist, to our knowledge none of the methods were trained and tested on GRID corpus. However, results on several lip-reading systems (visual only inputs) on GRID corpus have been reported. Table 2 gives a comparison of lip-reading systems which employ recurrent neural networks. LipNet is a recent independent work which uses spatio-temporal convolutions and Gated Recurrent Units. It is

Table 1. % CER comparison for feature fusion (RNN_{av}) and decision fusion (RNN_a, RNN_v) models. RNN_a is the acoustic model and RNN_v is the lip reader.

Feature fusion				Decision fusion			
Model	Input		CER %	Model	Input		CER %
	Audio	Visual			Audio	Visual	
RNN_{av}	Clean	OFF	7.35	RNN_a, RNN_v	Clean	OFF	2.45
RNN_{av}	Clean	ON	5.74	RNN_a, RNN_v	Clean	ON	8.46
RNN_{av}	OFF	ON	11.42	RNN_a, RNN_v	OFF	ON	11.06
RNN_{av}	10 SNR dB	OFF	38.31	RNN_a, RNN_v	10 SNR dB	OFF	23.83
RNN_{av}	10 SNR dB	ON	10.24	RNN_a, RNN_v	10 SNR dB	ON	14.83
RNN_{av}	0 SNR dB	OFF	59.65	RNN_a, RNN_v	0 SNR dB	OFF	59.27
RNN_{av}	0 SNR dB	ON	11.57	RNN_a, RNN_v	0 SNR dB	ON	16.84

trained using CTC at sentence level like our model whereas the RNN-LSTM model in [27] is trained at word level. However, in contrast to LipNet our aim in this paper was to present a noise-robust ASR which utilizes both audio and visual modalities which we believe will perform better with larger vocabulary datasets. Our model has the potential to switch from audio to a mixed modality (by turning the camera on) based on an SNR measure (where we define the signal as a continually discernible linguistic content from an utterance as measured perhaps using KL divergence described before). The %CER for LipNet [1] and the RNN-LSTM model of Wand et al., [27] are reported from [1].

Table 2. % CER comparison of lip-reading systems employing RNNs. The audio modality for the model in the last row is turned off.

Method	CER %
LipNet	1.90
Wand et al.	15.20
RNN_v	11.06
RNN_{av}	11.42

7 Conclusions and Future Work

In this work we presented an audio-visual ASR system using deep RNNs trained with CTC objective function. We described a feature processing step for visual features using deep bottleneck layer and showed that it helps in faster convergence of RNN model during training. We presented a training protocol in which either of the modalities is turned off during training in order to avoid dependency on a single modality. Our results indicate that the trained model is robust to noise. In addition, we compared fusion strategies at the feature level and at the decision level.

While the use of bottleneck features for visual modality helps in training, it requires frame level labels which involves an additional step of training audio RNN. Therefore, our system is not yet end-to-end. Our experiments in visual feature engineering with unsupervised methods like multi-modal auto-encoder [22] did not produce remarkable results. Currently, we are exploring visual features like curl and divergence of optical flow field using the Fourier Transform based on Clifford Algebra [6,20]. In future work we intend to explore other unsupervised methods for visual feature extraction such as canonical correlation analysis.

References

1. Assael, Y.M., Shillingford, B., Whiteson, S., de Freitas, N.: Lipnet: Sentence-level lipreading. arXiv preprint arXiv:1611.01599 (2016)
2. Bengio, S.: Multimodal speech processing using asynchronous hidden markov models. Inform. Fusion **5**(2), 81–89 (2004)
3. Brand, M., Oliver, N., Pentland, A.: Coupled hidden markov models for complex action recognition. In: Proceedings IEEE Computer Society Conference on Computer vision and pattern recognition, pp. 994–999. IEEE (1997)
4. Cooke, M., Barker, J., Cunningham, S., Shao, X.: An audio-visual corpus for speech perception and automatic speech recognition. J. Acoust. Soc. Am. **120**(5), 2421–2424 (2006)
5. Dupont, S., Luettin, J.: Audio-visual speech modeling for continuous speech recognition. IEEE Trans. Multimedia **2**(3), 141–151 (2000)
6. Ebling, J., Scheuermann, G.: Clifford fourier transform on vector fields. IEEE Trans. Vis. Comput. Graph. **11**(4), 469–479 (2005)
7. Gehring, J., Miao, Y., Metze, F., Waibel, A.: Extracting deep bottleneck features using stacked auto-encoders. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3377–3381. IEEE (2013)
8. Graves, A.: Supervised Sequence Labelling with Recurrent Neural Networks. SCI, vol. 385, pp. 15–35. Springer, Heidelberg (2012)
9. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376. ACM (2006)
10. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In: ICML, vol. 14, pp. 1764–1772 (2014)
11. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al.: Deep speech: scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567 (2014)
12. Hermansky, H., Ellis, D.P., Sharma, S.: Tandem connectionist feature extraction for conventional hmm systems. In: Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2000, vol. 3, pp. 1635–1638. IEEE (2000)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Huang, J., Kingsbury, B.: Audio-visual deep learning for noise robust speech recognition. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7596–7599. IEEE (2013)

15. Huang, J., Potamianos, G., Neti, C.: Improving audio-visual speech recognition with an infrared headset. In: AVSP 2003-International Conference on Audio-Visual Speech Processing (2003)
16. Katsaggelos, A.K., Bahaadini, S., Molina, R.: Audiovisual fusion: challenges and new approaches. Proc. IEEE **103**(9), 1635–1653 (2015)
17. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867–1874 (2014)
18. Miao, Y.: Kaldi+pdnn: building dnn-based asr systems with kaldi and pdnn. arXiv preprint arXiv:1401.6984 (2014)
19. Miao, Y., Gowayyed, M., Metze, F.: Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 167–174. IEEE (2015)
20. Mohammadzade, H., Bruton, L.T.: A simultaneous div-curl 2D clifford fourier transform filter for enhancing vortices, sinks and sources in sampled 2D vector field images. In: IEEE International Symposium on Circuits and Systems, ISCAS 2007, pp. 821–824. IEEE (2007)
21. Mroueh, Y., Marcheret, E., Goel, V.: Deep multimodal learning for audio-visual speech recognition. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2130–2134. IEEE (2015)
22. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 689–696 (2011)
23. Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H.G., Ogata, T.: Audio-visual speech recognition using deep learning. Appl. Intell. **42**(4), 722–737 (2015)
24. Potamianos, G., Neti, C., Gravier, G., Garg, A., Senior, A.W.: Recent advances in the automatic recognition of audiovisual speech. Proc. IEEE **91**(9), 1306–1326 (2003)
25. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al.: The kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. No. EPFL-CONF-192584. IEEE Signal Processing Society (2011)
26. Veselỳ, K., Ghoshal, A., Burget, L., Povey, D.: Sequence-discriminative training of deep neural networks. In: INTERSPEECH, pp. 2345–2349 (2013)
27. Wand, M., Koutník, J., Schmidhuber, J.: Lipreading with long short-term memory. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6115–6119. IEEE (2016)
28. Yu, D., Seltzer, M.L.: Improved bottleneck features using pretrained deep neural networks. In: Interspeech, vol. 237, p. 240 (2011)