END-TO-END LOW-RESOURCE LIP-READING WITH MAXOUT CNN AND LSTM

Ivan Fung and *Brian Mak*

The Hong Kong University of Science and Technology
Department of Computer Science and Engineering
Hong Kong
{hlfungaa, mak}@cse.ust.hk

ABSTRACT

Lip-reading is the task of recognizing speech solely from the visual movement of the mouth. Although recent works have demonstrated the effectiveness of convolutional neural network (CNN) and long short-term memory (LSTM) recurrent neural network in lip-reading, similar architectures under low-resource scenario have not yet been explored. Our proposed end-to-end deep learning model fuses conventional CNN and bidirectional LSTM (BLSTM) together with maxout activation units (maxout-CNN-BLSTM), and is capable of attaining a word accuracy of 87.6% on the Ouluvs2 corpus, offering an absolute improvement of 3.1% to the previous state-of-the-art auto-encoder-BLSTM model. To the best of our knowledge, this is the first end-to-end low-resource lip-reading system that does not require any separate feature extraction stage nor pre-training phase with external data resources. This is also the first work that utilizes maxout units in both CNN and LSTM in one single deep neural network.

Index Terms**—** lip-reading, visual speech recognition, low-resource, end-to-end deep learning, maxout activation

1. INTRODUCTION

Visual speech recognition (a.k.a. lip-reading) is the technology of interpreting speech through mouth movement without any audio aid. Whilst this is crucial for the hearing impaired to understand speech, it is also natural for the others to employ this technique to help determine speech in situations where audio alone is ambiguous, especially under noise-corrupted or far-field scenarios.

With the recent advancement in computational power, now it becomes feasible to accomplish this task through a wide spectrum of machine learning approaches, from latent variable models to hidden Markov models and artificial neural networks. Many of them have shown decent performance over various corpora, including the high-resource Lip Reading in the Wild (LRW) [1] and low-resource Ouluvs2 [2]. In this paper, we are going to present a pure end-to-end deep neural network that makes use of convolutional neural network (CNN) and long short-term memory (LSTM) recurrent neural network with maxout activation units for the low-resource Ouluvs2 lip-reading task. Unlike the previous work [3], we are able to obtain superior results on the task with this architecture without any additional training resources.

2. LITERATURE REVIEW

Conventional methods on lip-reading include latent variable models [4] and hidden Markov models (HMM) [5, 6], in which a prior separate stage of feature extraction is required. With the growing popularity of artificial neural network models, deep belief networks (DBN) such as auto-encoder and restricted Boltzmann machine (RBM) have been used as feature extractors, in conjunction with support vector machines (SVM) as the label classifiers [7, 8]. Recently, there is a neural network connecting encoding layers from a separate RBM pre-trained auto-encoder to LSTM layers, followed by an end-to-end training stage [3]. This model is able to reach an accuracy of 84.5% in the low-resource lip-reading Ouluvs2 corpus, which is the current state-of-the-art result without recourse to external training data.

On the other hand, numerous models pre-trained with extra data resources succeed in getting better accuracies in low-resource lip-reading tasks over the aforementioned models. For example, a frame concatenated model [9] that uses a number of deep pre-trained CNNs such as GoogLeNet [10] is able to achieve a better accuracy of 85.6% on Ouluvs2. Another work [1] exploiting multi-tower 3D convolutional neural network (3D CNN) that resembles the very deep convolutional network (VGG) [11] and multiple layer perceptron (MLP) further improves the accuracy to 93.2%, in which a separate pre-training stage using the very large LRW corpus is required. However, as far as we know, experiments of end-to-end networks combining CNN with LSTM without external data on low-resource corpora such as Ouluvs2 have not yet been reported. Moreover, leveraging the power of maxout activation units [12] in both CNN and LSTM in one single deep neural network has not been attempted before.

3. MAXOUT NETWORK

Maxout unit is a simple yet elegant activation function that is believed to work better in combination with dropout owing to its more accurate approximate model averaging capability [12]. It is proposed as a plausible choice for replacing ReLU, which is criticized for its high saturation rate at zero, and as an alternative to ReLU's other improved versions such as leaky [13] and randomized ReLUs. For a neural network with the previous hidden layer of size d, the current hidden layer of size m, and a number of k feature maps, the output of the i-th node of the current hidden layer, denoted as $h_i(\mathbf{x})$, can be characterized by the following simple formula:

$$h_i(\mathbf{x}) = \max_{j \in [1,k]} \{\mathbf{x}^T \mathbf{W}_{\cdot ij} + b_{ij}\} \,, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{W} \in \mathbb{R}^{d \times m \times k}$ and $\mathbf{b} \in \mathbb{R}^{m \times k}$.

Whilst maxout units for CNN (maxout-CNN) would be equivalent to max-pooling across channels with stride equal to k, LSTM can incorporate maxout by replacing the hyperbolic tangent activation in the memory gate, resulting in the following maxout-LSTM:

$$\begin{aligned}
i_t &= \sigma(\mathbf{W}_i \cdot [C_{t-1}, \mathbf{h}_{t-1}, \mathbf{x}_t] + b_i) &(2)\\
f_t &= \sigma(\mathbf{W}_f \cdot [C_{t-1}, \mathbf{h}_{t-1}, \mathbf{x}_t] + b_f) &(3)\\
o_t &= \sigma(\mathbf{W}_o \cdot [C_{t-1}, \mathbf{h}_{t-1}, \mathbf{x}_t] + b_o) &(4)\\
\tilde{C}_t &= \max_{j \in [1,k]} \{\mathbf{W}_{Cj} \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + b_{Cj}\} &(5)\\
C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t &(6)\\
h_t &= o_t \cdot \tanh(C_t) &(7)
\end{aligned}$$

We notice that another work has also utilized maxout activation units in the LSTM alone [14].

4. OULUVS2 CORPUS

We chose Ouluvs2 because it is a low-resource corpus consisting of a reasonable number of disparate subjects for training. The corpus is composed of 3 distinct parts, 5 views and 52 subjects, of which 39 are male and 13 are female. The first part of the corpus is a set of 10 different strings of digits from 0 to 9 in random order, while the second part is 10 different daily-use short phrases, and the third part is 10 random sentences adopted from the TIMIT corpus [15]. Similar to the previous work [3], we used only the frontal view of the second part of the corpus in our evaluation section, which comprises the following 10 phrases: 'Excuse me', 'Goodbye', 'Hello', 'How are you', 'Nice to meet you', 'See you', 'I am sorry', 'Thank you', 'Have a good time', and 'You are welcome'.

Amongst the 10 distinct phrases, every subject repeats each of them 3 times; therefore, a total number of 156 samples are provided for each phrase, as compared with 800-1100 samples for each word in the high-resource LRW corpus. Following the traditional data splitting scheme as suggested by the author [9], we reserved subjects 06, 08, 09, 15, 26, 30, 34, 43, 44, 49, 51 and 52 for testing, in which 10 of them are male and 2 of them are female, and the remaining for training.

5. EXPERIMENTS

We present our experiments in three parts, namely data preprocessing, network architecture and hyper-parameters, and evaluation results.

5.1. Data preprocessing

Each video clip was converted to a sequence of lossless grayscale images of variable length using the FFmpeg [16] software. Then we performed a 1:2 crop around the mouth region found by dlib [17] and contrast enhancement in each image. Afterwards, each image was downsampled to 16×32 using bicubic interpolation smoother. Data augmentation was carried out by shifting the cropping area to 8 different directions (top-left, top, top-right, right, bottom-right, bottom, bottom-left, and left) by 10 pixels, followed by a further augmentation through horizontal flipping, to create a total of 16 different copies from the original image. Finally, each image was z-normalized across each pixel through a global mean subtraction and variance normalization. The whole preprocessing procedure is depicted in Fig. 1.

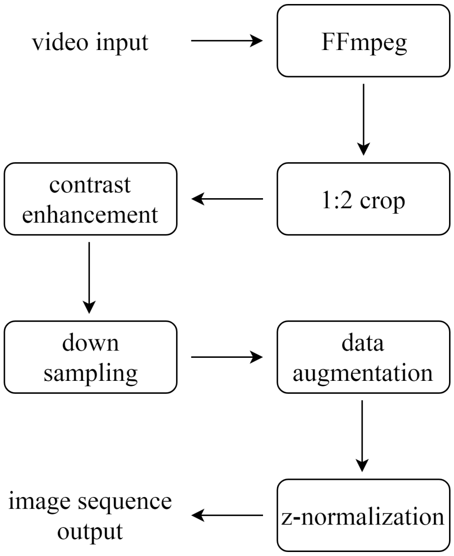

Fig. 1. Preprocessing flow-chart

5.2. Network architecture and hyper-parameters

Our end-to-end deep neural network is comprised of two parts. The first part contains 8 layers of convolutional layers as the visual front-end and the second part contains one layer of bidirectional LSTM (BLSTM) as the sequence learning

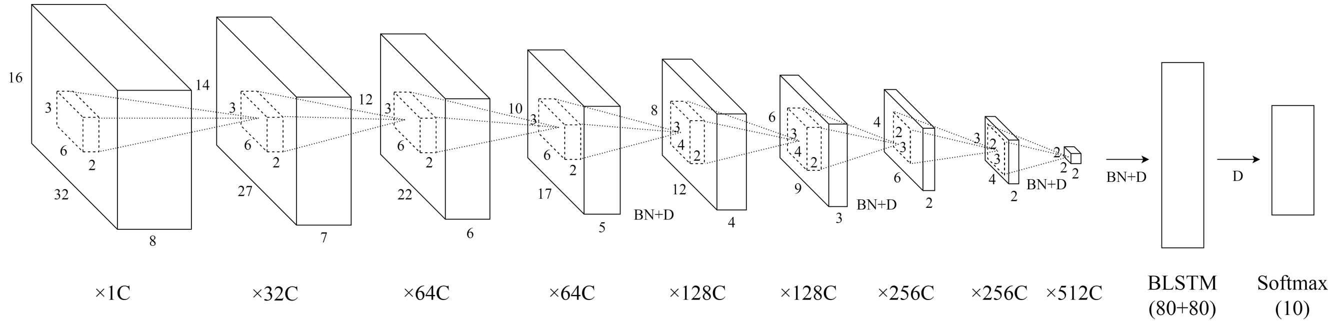

Fig. 2. Network architecture of the maxout-CNN-BLSTM model. C: Channel; BN: Batch Normalization; D: Dropout.

back-end. Each of the convolutional layers is a spatial-temporal convolution (3D convolution) with no zero-padding or stride, followed by an activation function, either a maxout or ReLU unit, without any pooling layer. For the BLSTM layer, either the common bidirectional peephole LSTM using the hyperbolic tangent activation, or its maxout version described in Section 3 was used. Finally, outputs of the forward and backward LSTM of the last frame of each input sequence were concatenated together into a vector, which serves as the input to the softmax classification layer of 10 targets.

In order to demonstrate the effectiveness of the maxout activation units in the deep neural network, we carried out the experiment under four different setups, namely ReLU-CNN with tanh-BLSTM, ReLU-CNN with maxout-BLSTM, maxout-CNN with tanh-BLSTM, and maxout-CNN with maxout-BLSTM (maxout-CNN-BLSTM) respectively. Note that the input at each time step is a stack of 8 consecutive images obtained from a sliding window along the image sequence, i.e. a tensor of $16 \times 32 \times 8$. Fig. 2 gives the network architecture of the maxout-CNN-BLSTM as an example.

To alleviate the problem of overfitting, we employed a number of regularization methods including batch normalization, dropout and L2-regularization. Whilst batch normalization layer was inserted between various layers in CNN, a dropout rate of 0.5 was applied to the whole network starting from the 4th epoch, and an L2-regularization with weight 0.00155 was applied to all trainable parameters to penalize highly positive and negative values. Along with batch normalization, a momentum of 0.6 was used in the first 10 epochs followed by 0.9 in the remaining epochs to speed up convergence in the training stage. Initial learning rate was set to 0.01 and was reduced by roughly half after every 2 epochs. A mini-batch size of 256 images, not image sequences, was used, and a total number of 15 epochs were run in every setup.

5.3. Evaluation results

We implemented and evaluated our models using CNTK [18], which takes great advantage of the parallel computations in GPUs to improve training speed. To improve performance reliability, each of the above experimental setups was repeated 12 times. During each run, the training set of 40 subjects was randomly partitioned into two non-overlapping groups of 4 and 36 subjects respectively. The small and large partitions were used as the validation and training data respectively. The reported result of each setup is the average of testing accuracies under 12 respective runs, where each was evaluated on the epoch with the lowest validation error.

Table 1. Classification accuracy of various models.

Method ($k = 4$ for maxout)	Accuracy (%)
Auto-encoder with tanh-BLSTM [3]	84.5
ReLU-CNN with tanh-BLSTM	84.6
ReLU-CNN with maxout-BLSTM	84.4
maxout-CNN with tanh-BLSTM	85.6
maxout-CNN-BLSTM	87.6

It can be seen from Table. 1 that our proposed maxout-CNN-BLSTM model is the best among the tested models and is able to obtain a state-of-the-art accuracy of 87.6% in the low-resource Ouluvs2 task without resorting to any other external data resources. This also confirms the superior performance of maxout unit over the conventional ReLU and tanh in deep neural network, probably because it is free of the high zero saturation rate problem that occurs in ReLU, and has more accurate approximate model averaging with dropout.

Table 2. Training time (hr) of various models (each run).

Method ($k = 4$ for maxout)	Time (hr)
ReLU-CNN with tanh-BLSTM	2.4
ReLU-CNN with maxout-BLSTM	2.5
maxout-CNN with tanh-BLSTM	7.8
maxout-CNN-BLSTM	7.8

From Table. 2, it can be seen that CNN with maxout units increases the training time to more than 3 times to that with ReLU. This confirms the use of maxout units involves a k-time increase in the network parameter size, which in turn leads to many more computations. On the other hand, the difference in training time between BLSTM with hyperbolic tangent activations and that with maxout units is minor. Nonetheless, maxout units are still beneficial due to the abovementioned accuracy gain.

Table 3. Effect of various number of maxout feature maps, k.

maxout-CNN-BLSTM	Accuracy (%)	Time (hr)
$k = 2$	85.6	4.2
$k = 3$	86.1	6.2
$k = 4$	87.6	7.8
$k = 5$	86.3	10.0

To further investigate the maxout activation units, we have conducted experiments on the effect of the number of feature maps, denoted as k, in the maxout-CNN-BLSTM architecture. As shown in Table. 3, we observe that the time of computation increases with the number of feature maps, and $k = 4$ offers a slightly better accuracy in comparison with others. It also confirms that even with only two feature maps, it is already sufficient to approximate arbitrarily sophisticated and non-linear functions, having a similar effect to other activation functions such as ReLU and tanh.

6. DISCUSSION

In this section, we will make some key comparisons to the auto-encoder-BLSTM model, and explain the difficulties in coming up with our final maxout-CNN-BLSTM architecture.

6.1. Comparisons to auto-encoder-BLSTM

We propose using a CNN as a replacement of the auto-encoder employed in the previous work [3] chiefly because of its capability of capturing spatial correlations. We believe that a CNN, of which each convolutional layer is designed and intended to work as a filter to capture local correlations along the spatial dimensions, will not work worse than encoding layers in an auto-encoder.

Using a CNN front-end also allows us to extend the 2D convolution (using 2D filters) to 3D convolution (using 3D filters) by taking into account the additional temporal dimension so as to capture the temporal correlations among successive images on top of the spatial correlations in an image. Results show that 3D convolution can provide a substantial gain in the lip-reading performance. In [3], the authors had trained encoder layers on the differences between images, which should also model the temporal correlations among successive images to some extent, contributing to a significant performance gain in their work as well. However, we believe that the use of 3D convolution in our work is more effective.

One may question about the use of 3D convolution in the front-end given that the back-end LSTM can also learn the temporal dependency among the images in the input image sequence. Our results show that it is better to do both. We believe that the 3D convolutions performed in the front-end can probably capture the short-term temporal correlations among the successive images, and the resulting feature maps can thus provide the back-end LSTM with a more global view of the image sequence to help to capture both the long-term and short-term correlations from the image sequence.

6.2. Difficulties in training with CNN-BLSTM

The previous work [3] failed to reach better accuracy using a CNN as the visual feature extractor when compared with our work, probably due to multiple reasons in its network design and training strategies. First and foremost, we used convolutional layers to reduce each stacked image input to a small size before feeding it to the BLSTM. That is, in our case, it is 2×2×2 along width, height and temporal depth of the images. We found that any dimension above 2 would lead to worse performance. Second, the maxout activation works better in comparison with the conventional ReLU activation in CNN and tanh activation in BLSTM. As demonstrated by the maxout-CNN-BLSTM architecture, the maxout activation provides a considerable absolute gain of 3.0% in accuracy compared to its counterparts. Third, techniques of preventing overfitting are important across the whole network. Among the aforementioned three methods, L2-regularization has the most direct impact in addressing this problem. It can prevent the network weights from becoming too positive or negative. Finally, data augmentation is important for training such a deep network for a low-resource corpus. Though a deep neural network is well-known for its ability in learning high-level and abstract features, that happens only when a sufficient number of training samples is provided.

7. CONCLUSION

We have successfully demonstrated the capability and feasibility of designing an end-to-end deep neural network for the low-resource lip-reading task using CNN and BLSTM with incorporation of maxout activation units. We are able to achieve a state-of-the-art accuracy of 87.6% on the Ouluvs2 10-phrase task without using any external data resources. In the future, we are going to explore the possibility of applying the maxout units in larger and more difficult lip-reading tasks such as the visual speech recognition of sentences [19, 20], through utilizing other end-to-end architectures.

8. ACKNOWLEDGEMENTS

The work described in this paper was partially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. HKUST16206714 and HKUST16215816).

9. REFERENCES

[1] J. S. Chung and A. Zisserman, “Lip reading in the wild,” in *Proceedings of the Asian Conference on Computer Vision*, 2016.

[2] I. Anina, Z. Zhou, G. Zhao, and M. Pietikäinen, “Ouluvs2: a multi-view audiovisual database for non-rigid mouth motion analysis,” in *Proceedings of the International Conference and Workshops on Automatic Face and Gesture Recognition*. IEEE, 2015, vol. 1, pp. 1–5.

[3] S. Petridis, Z. Li, and M. Pantic, “End-to-end visual speech recognition with LSTMs,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017, pp. 2592–2596.

[4] Z. Zhou, X. Hong, G. Zhao, and M. Pietikinen, “A compact representation of visual speech data using latent variables,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, 2014.

[5] G. I. Chiou and J. N. Hwang, “Lipreading from color video,” *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1192–1195, 1997.

[6] S. Dupont and J. Luettin, “Audio-visual speech modeling for continuous speech recognition,” *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 141–151, 2000.

[7] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. L. Lee, and A. Y. Ng, “Multimodal deep learning,” in *Proceedings of the International Conference on Machine Learning*, 2011, pp. 689–696.

[8] N. Srivastava and R. R. Salakhutdinov, “Multimodal learning with deep Boltzmann machines,” in *Advances in Neural Information Processing Systems*, 2012, pp. 2222–2230.

[9] T. Saitoh, Z. Zhou, G. Zhao, and M. Pietikäinen, “Concatenated frame image based CNN for visual speech recognition,” in *Proceedings of the Asian Conference on Computer Vision*, 2016, pp. 277–289.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[11] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in *Proceedings of the British Machine Vision Conference*, 2014.

[12] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, “Maxout networks,” in *Proceedings of the International Conference on Machine Learning*, 2013, vol. 28.

[13] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proceedings of the International Conference on Machine Learning*, 2013, vol. 30.

[14] X. G. Li and X. H. Wu, “Improving long short-term memory networks using maxout units for large vocabulary speech recognition,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2015, pp. 4600–4604.

[15] W. M. Fisher, “The DARPA speech recognition research database: specifications and status,” in *Proceedings of the DARPA Workshop Speech Recognition*, 1986, pp. 93–99.

[16] FFmpeg team, “FFmpeg,” https://ffmpeg.org.

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

[18] F. Seide and A. Agarwal, “CNTK: Microsoft’s open-source deep-learning toolkit,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 2135–2135.

[19] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, “Lip reading sentences in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[20] M. Cooke, J. Barker, S. Cunningham, and X. Shao, “An audio-visual corpus for speech perception and automatic speech recognition,” *Journal of Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.