End-to-End Phoneme Recognition using Models from Semantic Image Segmentation

Wei Gao*, Ahmad Hashemi-Sakhtsari† and Mark D. McDonnell*
*Computational Learning Systems Laboratory,
School of Information Technology and Mathematical Sciences,
University of South Australia, Mawson Lakes SA 5095, Australia
Email: gaowy009@mymail.unisa.edu.au, mark.mcdonnell@unisa.edu.au

†Language Technology and Cognition Group,
Intelligence, Surveillance and Space Division,
Defence Science and Technology Group, Edinburgh SA 5111, Australia
Email: ahmad.hashemi-sakhtsari@dst.defence.gov.au

***Abstract*—We train fully convolutional neural networks with no recurrent layers for the end-to-end phoneme recognition task, using the Connectionist Temporal Classification (CTC) loss function. The adopted network, U-Net, was introduced initially for semantic image segmentation tasks, and is often applied to segmenting features in medical imaging and remote sensing. The similarities between CTC-based automatic speech recognition and semantic segmentation problems are discussed. We extend the encoder-decoder architecture of U-Net and show it is capable of good performance in the acoustic modelling of a speech recognition system. We investigate the importance of the concatenation step in the design of U-net, and report results using the core test set of the TIMIT corpus.**

***Index Terms*—speech recognition, semantic image segmentation, convolutional neural networks, connectionist temporal classification**

I. Introduction

Conventional automatic speech recognition systems (ASR) consist of multiple modules that are independently designed and optimised. Prior studies have made progress to reduce the intrinsic complexities of building such a highly integrated system, e.g. [1]–[3]. The most intuitive and widely investigated approach is to develop end-to-end ASR systems, which attempt to transform acoustic features directly into word-level representations with fewer or no intermediate components.

Prior to the emergence of end-to-end systems, the typical solution for ASR relied on separate components for feature extraction, acoustic models, language models and pronunciation dictionaries (lexicon) [4]. Weighted finite-state transducers (WFST) were commonly deployed to combine the representation of each individual sub-system, and to implement a weight conveying strategy that defined the mapping from the inputs to the outputs [5]. The feasibility of building end-to-end ASR systems is attributed to the application of deep neural networks (DNN) in acoustic modeling, which now outperform the previous solution: Gaussian Mixture Models with hidden Markov Models (GMM-HMM) [4], [6]. DNN-HMM hybrid models were shown to better exploit the contextual information behind adjacent acoustic frames via more accurate feature representation and improved discriminability power in various ASR tasks [7], [8]. Later, the applications of convolutional neural networks (CNN) and recurrent neural networks (RNN) in acoustic modelling led to new state-of-art results [2], [9]–[11]. Despite this progress, the resulting hybrid models still require highly specialised inputs such as Mel-filterbank cepstral coefficients (MFCCs) to succeed. A complicated system design is also essential in order to integrate independent modules, with subsequent fine-tuning to enhance performance, which however makes the optimisation difficult. There is room for improvement when it comes to feature pre-processing and systematic design for ASR.

Based on the success of DNN-based acoustic modelling, end-to-end ASR systems handle both feature classification and sequence decoding without knowing the alignments between the acoustic features and the transcription. There are two major end-to-end architectures that attempt to build direct mappings between acoustic frames and predictable tokens: sequence-to-sequence models with an attention mechanism [3], [12], and DNNs trained using the Connectionist Temporal Classification (CTC) objective function [1], [2]. As recently reviewed in [13], training acoustic models in an end-to-end fashion using CTC can reach higher accuracy than training traditional hybrid-CTC models while attention-based models have not yet outperformed its traditional counterpart due to computational complexity. Therefore, we shall explore CTC-based solutions for acoustic modeling in this paper.

We propose that CTC-based ASR bears a resemblance to semantic image segmentation tasks. Semantic segmentation of an image means to classify every pixel into one of N categories. State of the art semantic segmentation uses supervised learning and deep convolutional neural networks to predict the segmentation boundaries and the object enclosed.

End-to-end ASR models trained using CTC loss aim to build a mapping from acoustic sequences to predictable to-

This work was funded by Scholarship (Project-Based) Funding Agreement with Australia's Defence Science and Technology Group (DSTG), and an Australian Government Research Training Program (RTP) Scholarship.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

kens (e.g. phonemes, characters and words) for each frame of speech. The segmentation boundaries of different phonemes appearing in the input sequence are automatically learned given the repeated predictions made between neighbouring acoustic frames. The repeated tokens are further pruned when decoding the CTC loss in order to ensure a unique classification result across several frames. Although this two-step process of handling CTC loss in ASR is conceptually similar to semantic image segmentation, an important difference is that the ground truth for semantic segmentation of images requires annotated masks formed from manually assigning a class label to each pixel of the image. This is different to transcriptions used for training ASR where the alignment between speech frames and transcript tokens is usually unknown.

Networks for semantic segmentation, such as models that employ spatial pyramid pooling, dilated convolutions, and designs using an encoder-decoder architecture, as in U-net, have shown strong capabilities in exploiting contextual interaction among pixels [14]–[16]. This is consistent with the advantage of training CNNs with CTC loss for end-to-end ASR models. We chose U-net for this study since it has shown to be effective in solving semantic segmentation of images with very limited training data [16].

II. Methods

A. Review of CTC loss

The CTC function takes the output of softmax layers as input and generates the distribution probability of each symbol for each frame in an input sequence [1]. A blank token ϕ is introduced to force the input sequence and the CTC output to remain the same length, and to indicate where to merge repeated frames assigned to the same category. Consider $\boldsymbol{l}$ as the target input label sequence and $\boldsymbol{B^{-1}(l)}$ as the mapping from all possible connecting paths to the target sequence, for example, $\boldsymbol{B}(A\phi BB\phi\phi C) = \boldsymbol{B}(\phi AA\phi BC) = ABC$. Any repeated tokens and blanks are accordingly removed. The CTC loss $P(\boldsymbol{l}|\boldsymbol{x})$ can be calculated by adding the probabilities of all potential paths given the input sequence:

$$P(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\pi \in \boldsymbol{B^{-1}(l)}} P(\pi|\boldsymbol{x}), \tag{1}$$

where x denotes the input acoustic sequence $\boldsymbol{x} = [x_1, x_2, ..., x_T]$ with length T, and $P(\pi|\boldsymbol{x})$ denotes the probability of one single path π given the input sequence, defined by

$$P(\pi|\boldsymbol{x}) = \prod_{t=1}^{T} y_k^t, \tag{2}$$

where y_k^t is the output of softmax activation, representing the probability of label k over $k+1$ classes at time step t.

Dynamic programming is used to efficiently calculate the CTC loss. For inference after training, we can find the most likely label sequence by solving

$$\boldsymbol{h}(x) = \underset{\boldsymbol{l}}{\operatorname{argmax}} P(\boldsymbol{l}|\boldsymbol{x}). \tag{3}$$

We chose the beam search algorithm with a beam width of 10 to construct the most possible sequence in terms of one-hot encoded frame-wise predictions.

B. U-Net architecture

U-Nets are fully convolutional networks in an encoder-decoder architecture, without any recurrent layers [16]. The resulting intermediate features processed by the encoder with large receptive fields can learn global spatial invariance, which in our case becomes frequency-time patterns. Concatenations aggregating the feature maps at different scales are designed to ensure the decoder can use both local information from those layers close to input, and global information from layers closer to the output. We hypothesise that acoustic modeling for ASR can benefit from such an architecture as both local information and contextual information are as important in recognising frequency-time patterns in speech as they are for segmenting objects in images.

Fig.1 illustrates our proposed adaptation of a U-net architecture, that serves as the baseline model in our experiments. The encoder is constructed by stacking two subsequent blocks consisting of batch normalisation layers [17], pre-activation with ReLU layers and 2D convolution layers with a kernel size of 3×3 to produce the feature maps. Dropout is also applied to help alleviate overfitting. Then, downsampling is performed following the end of each encoder block, by applying max pooling with stride 2 on the frequency axis, and the number of feature channels is doubled. We also apply downsampling on the time axis but only apply it once, with the first max pooling operation. This procedure helps to retain the spectral invariance while reducing the size of feature maps.

Each encoder block has a corresponding block in the expansive path where each new input to a block is constructed by generating the concatenation of upsampled feature maps from the previous layer and the feature maps copied from the encoder. The number of feature channels is reduced through 3×3 convolutions within the block, identical to the number of channels from the copied feature maps. We apply weight decay with coefficient of 10^{-5} at each convolutional layer. Dropout with a rate of 0.2 is applied along the contracting path. All convolutional layers are initialised using He uniform scaling [18].

C. Experiment Details

1) Speech Corpus: TIMIT: We perform experiments on the TIMIT speech corpus which is a standard benchmark dataset for phoneme recognition tasks. We follow the strategy introduced in [19] to split the data, leading to 3,696 training utterances among 462 speakers with all dialect sentences (the sentences with *SA* tag) removed, and 400 sentences from 50 speakers as a validation set. For evaluation, we report phoneme error rates (PER) on the core test set, which consists of 192 sentences (from 24 speakers) out of the complete test

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

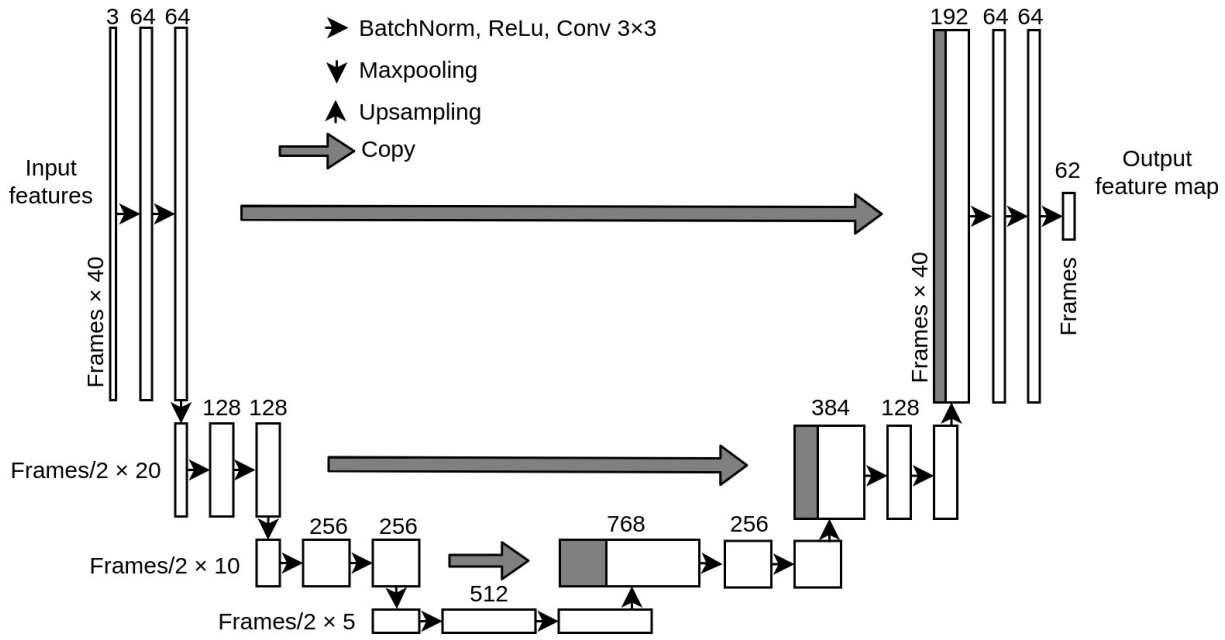

Fig. 1: Proposed U-Net architecture for acoustic modelling. The contracting path is at the left, the intermediate feature maps are located at the bottom, and the expansion path is at the right. Each white box represents a multi-channel filter for convolution. The number of feature channels is denoted on top of the box and the dimension of the feature map (in time frames and number of acoustic features) is also marked. Shaded boxes pointed to by thick grey arrows represent copied feature maps that are concatenated with feature maps in the expansion path. The small black arrows denote a set of operations which are commonly employed when training CNNs.

set [19], as well as the validation set. The target prediction labels are comprised of 61 English phonemes plus a blank token indicating necessary merging during both training and inference. The scoring is performed on 39 phonemes using a standard mapping from the original 61 phonemes [20]. We do not use time alignment information of TIMIT dataset, as it is not the common practice for building end-to-end ASR with CTC loss [2], [21].

At the stage of pre-processing, we compute mel-filter bank energy features and take the logarithm of all features, instead of mel frequency cepstral coefficients (MFCCs). Generating MFCCs previously was thought to be the best choice for feature extraction in the era of GMM-HMM based ASR, but the necessity of taking the discrete cosine transform in order to calculate MFCCs was questioned in DNN-based ASR given that DNNs prefer decorrelated features as inputs which exhibit more temporal and spatial smoothness in the case of ASR [22].

We compute 40-dimensional log mel-filter bank coefficients with its first and second derivatives at each 10ms of acoustic signal with a frame length of 25ms, resulting in 3 channels with a map of 40 frequency dimensions. Hamming window is applied after deploying short time Fourier transform. A portion of input features are padded by zero vectors along the frequency axis in some experiments. This is required to create inputs with a size equal to a multiple of two, since spatial pooling is applied when training U-net.

2) Training Setting: We train our models using a batch size of 24, with momentum of 0.9 and stochastic gradient descent (SGD). The learning rate schedule is adapted from a development-based decay scheme introduced by [23], where we monitor the validation performance at each epoch and reduced the learning rate by a multiplicative factor of 0.98 if worse validation performance is observed. The initial learning rate is set to 0.02, and the minimum learning rate allowed is 0.0001. As shown in Fig.1, the baseline model has 15 weight layers with filter size starting from 64. In total it has 7.8M trainable parameters.

III. Results and Discussion

Our work was inspired by the fully convolutional network architecture proposed for phoneme recognition [21], but the two models differ a lot in terms of input construction and network architecture. This is the only paper to our knowledge that studies end-to-end phoneme recognition models with no recurrent layer. The work presented in [21] reported a PER of 18.2% on the core test set. The TIMIT state-of-the-art result of 14.9% PER was achieved using a hybrid RNN/HMM design and a speaker adaptive training procedure [24], [25], which we consider as beyond the scope of our study.

Table I presents PER comparisons on the TIMIT dataset. Our best single model achieved a PER of 16.8% for the validation set and 18.8% for the test set. For all the results reported in this paper, we took the average value of PER from

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE I: PER comparisons in percentages on TIMIT development set and test set. The "conv layers" describes the architecture of the CNN-based models, in terms of the number of convolutional layers included in two symmetrical paths and the kernel size of each convolution layer. #filters indicates the number of output channels in the first convolutional layer. Subsequent layers increase the number of channels using the same factor in both contracting and expanding paths as explained in section 2.2. #params denotes the total number of trainable parameters. Regarding the concatenation step, the respective operations performed in contracting path and expansion path are included.

Conv layers (#layers-kernel-#filters)	#params	Details of concatenation step				Dev PER (%)	Test PER (%)
		Fig ID	Operations	Contracting path	Expansion path		
14 - 3,3 - 64	8.61M	a	concatenation	copy	conv(3,3)+upsampling	17.1	19.8
14 - 3,3 - 64	7.84M	b	concatenation	copy	upsampling	**17.0**	**19.6**
14 - 3,3 - 64	8.61M	c	sum	conv(3,3)	upsampling	17.7	20.1
14 - 3,3 - 64	8.61M	d	average	conv(3,3)	upsampling	17.6	20.1
18 - 3,3 - 32	7.87M	b	concatenation	copy	upsampling	18.3	20.8

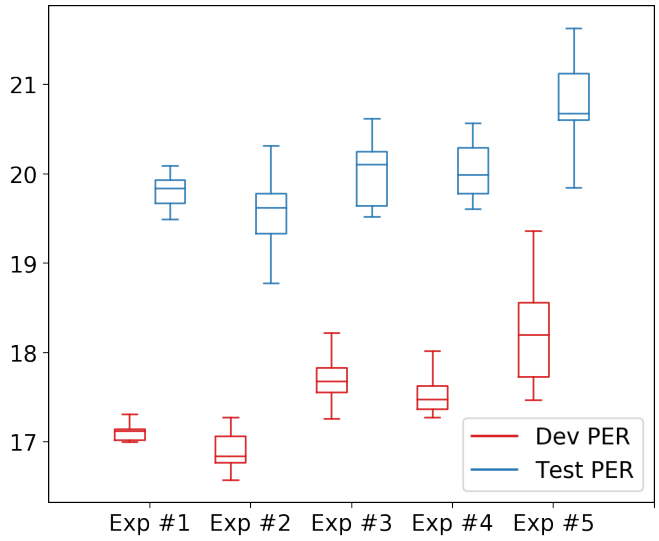

Fig. 2: PER comparison. The five horizontal bars from top to bottom respectively represent the maximum PER, the third quartile, the median PER, the first quartile, and the minimum PER, out of ten individual experiment results under same setting.

ten individual experiments. Fig. 2 shows the distribution of PERs across each experiment setting listed in Table I.

A. *Changes made to the concatenation step*

We made some modifications to build our baseline model which makes it different from the original U-Net architecture (Fig. 3a). We replaced the up-convolution with just upsampling in the expansion path before concatenations occurred (Fig. 3b). The purpose of this modification is twofold: first, U-nets with few convolutional layers had a smaller memory footprint during training and inference; second, it was expected to better retain the contextual information by not performing up-convolutions. The modified version of U-net showed better performance (average PER of 19.6%) compared to the performance with original design of U-net (average PER of 19.8%).

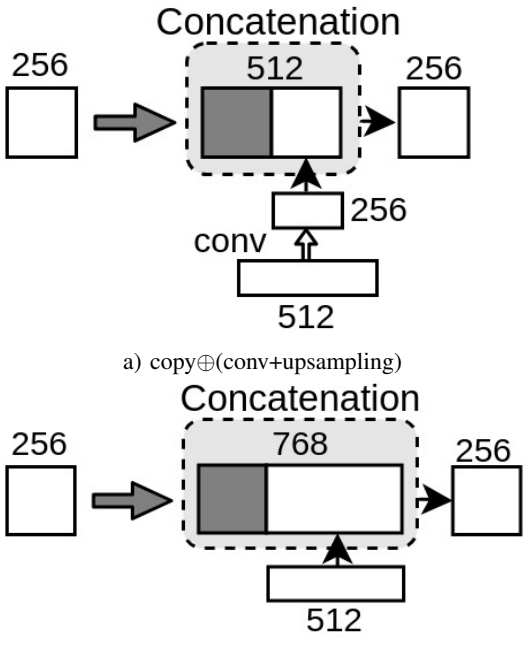

Fig. 3: Variations of concatenations step in U-net. These figures follow the same sign convention as explained in Fig.1.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE II: Comparison of the number of errors of each type from U-nets using different connection techniques and different windowing functions.

Error types	Model 3b (Rectangular window)	Model 3b (Hamming window)	Model 3c (Hamming window)
insertion	356 (25.3%)	176 (12.8%)	189 (13.2%)
deletion	202 (14.4%)	360 (26.1%)	374 (26.1%)
substitution	849 (60.4%)	841 (61.1%)	871 (60.7%)
total	1407 (PER 19.2%)	1377 (PER 18.8%)	1434 (PER 19.6%)

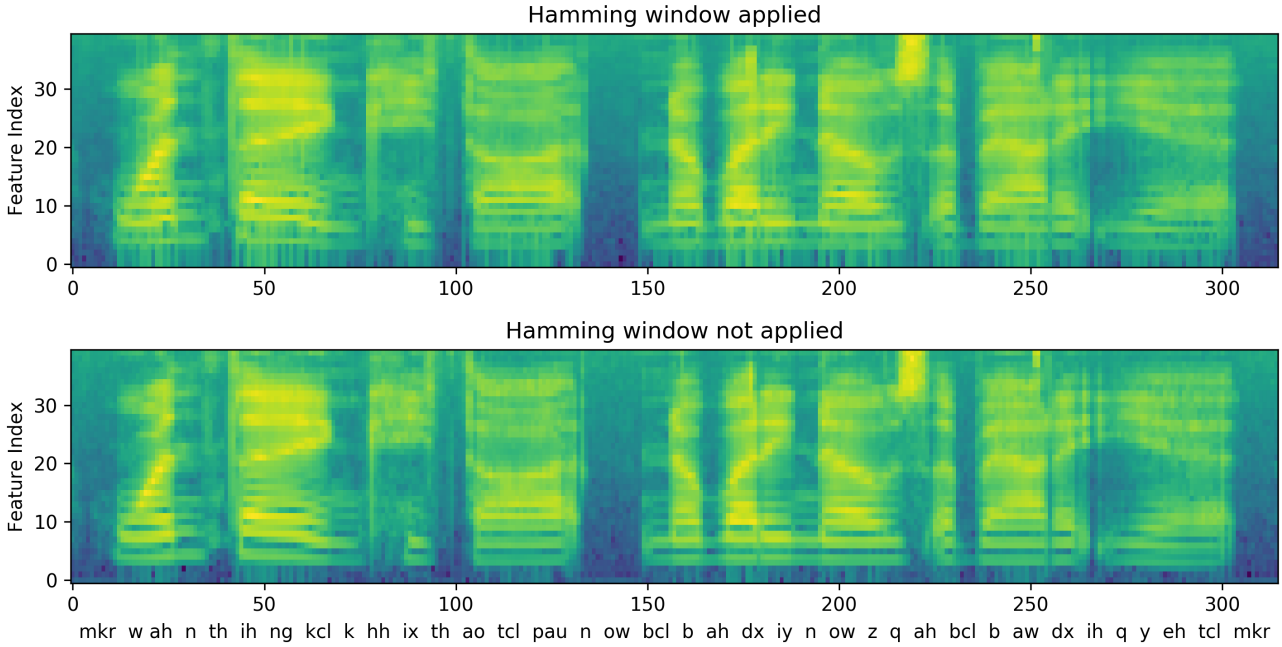

Fig. 4: Log mel-filter bank coefficients of one TIMIT utterance. The use of windowing funcition (up) brings variations in lower frequency band. The corresponding phonetic transcription is attached.

We also tried to apply convolutions on feature maps from the contracting path and double the number of filters. The dimension of transformed features from the contracting path is now identical to that of the upsampled feature maps. The new features to be processed in the expansion path is generated by performing pixel-wise addition or taking the average of these two feature maps (Fig. 3c & 3d) hence no concatenation is applied. A PER performance gap is observed in models not using concatenations compared to the baseline model.

B. Error analysis

There are 7,333 tokens from 192 sentences to be predicted in TIMIT core test set. Table II compares the number of incorrect predictions in terms of three error types and their portions of total misclassification from two models which differ on how features from the contracting paths are transformed (Fig. 3b & 3c). The PERs reported in this table are the best result out of ten individual experiments under the specific model setting. It is noted that the increase of each individual error contributes to relatively worse PER of Model 3c.

We also compare the performance of different choices for windowing function. We observed PER decline with Hamming window applied. In general, the number of insertion errors are largely reduced while the number of deletion errors was increased. The use of Hamming window helped to minimise the effects of Fourier Transform side lobes hence improved the overall recognition performance. Fig. 4 compares 40-dimensional log mel-filter bank coefficients with Hamming window applied from extracted acoustic features with Hamming window not applied.

The classification errors are explicitly investigated as shown by Table III and IV. We list five most frequent cases of each error type, sorted in descending order based on the error counts (#err) and the error counts over the true counts of each category in percentage (%err, false negative rate), except for the insertion errors where %err is calculated by dividing the error counts over the number of predictions of that particular category.

C. Impact of more convolutional layers

We firstly tried to train the U-Net with a deeper architecture, and hence another encoder block was stacked onto

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

TABLE III: Error analysis for the experiment using baseline model 3b from Fig. 3.

#err	true	predicted	%err	true	predicted
43	ah	ih	24.1	uh	ah
37	ih	ah	14.4	z	s
26	z	s	13.8	uh	ih
25	eh	ih	13.4	ah	ih
19	ih	iy	13.3	ah	eh
76	sil		20.0	y	
27	ih		10.4	hh	
28	r	(del)		r	(del)
22	n		10.3	uh	
20	ah		7.9	ow	
36		sil	6.5		v
19		ih	5.6		ow
16	(ins)	ah	5	(ins)	ah
11		r			ch
8		l	4.4		d

TABLE IV: Error analysis for the experiment using Model 3c from Fig. 3.

#err	true	predicted	%err	true	predicted
43	ah	ih	17.2	uh	ih
39	ih	ah	15.4	ng	n
24	ih	iy	15.2	ae	eh
21	eh	ih	13.8	uh	ah
	er	r	13.8	ah	ih
87	sil		22.0	y	
35	ih		14.3	hh	
27	r	(del)	12.5	oy	(del)
25	ah		10.3	uh	
24	l		10.0	r	
47		sil	8.8		d
20		ah	8		y
14	(ins)	ih	6.5	(ins)	hh
12		l			v
		r	6.2		ah

the contracting path with two more weight layers, and the expansion path accordingly made deeper to keep the symmetry of the network architecture. For comparison, we reduced the number of filters so as to train a network with similar scale to the baseline model.

We observed that the slightly higher PER as of 20.8% in average was mainly due to more insertion errors at both the beginning and the end of the predicted sequence. It is likely that the additional spatial paddings applied to the input leads to the problem. Those paddings are necessary to construct the input features with correct dimension in terms of subsequent pooling operations.

D. Data Augmentation

We tried to apply data augmentation, which is a common practice in image processing [26], but in all cases the test PER increased. For example, we unsuccessfully tried to take random crops from zero-padded input sequence. The outcome is intriguing because data augmentation routinely enables better performance in image classification, and our model constructs image-like inputs with spatial-temporal patters similar to objects in images. However, it should be noted that many forms of image data augmentation do not apply to our case because there is no parallel to flipping or reversing images, nor is there a need to search for similar patterns at different frequencies.

IV. Conclusion

We presented empirical experiments on training U-Nets on the TIMIT phoneme recognition task, showing it is reasonably capable of end-to-end phoneme recognition. Although an accuracy gap between the best results and ours remains, this study suggests it will be worthwhile to explore more powerful semantic segmentation models for speech applications.

As shown in [27], CTC-based models with word outputs significantly outperformed CTC-based models with phoneme output in terms of classification error rate given the identical data set. As for building ASR in end-to-end fashion, it is more practical to directly generate word sequences rather than phonemes from the acoustics. Hence, we suggest a priority in future work is to train U-nets with the CTC loss function in terms of word units. Then, it will also be interesting to introduce attention mechanisms to the proposed system, and compare the performance of fully-convolutional neural networks built under different end-to-end settings.

In addition, we considered the comments from reviewers that novel models based on the design of U-Net could be used for this task, hence we trained Residual U-Nets [28] but did not obtain improved results.

Acknowledgment

We gratefully acknowledge discussions with Dr. Ying Xu and Mr. Md Atiqul Islam from International Centre for Neuromorphic Systems of Western Sydney University.

References

[1] A. Graves, S. Fernández, F. Gomez, and J.Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[2] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[3] J. Chorowski, D. Bahdanau, K. Cho, and Y.Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: first results," in *NIPS 2014 Workshop on Deep Learning*, 2014.

[4] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," in *Foundation and Trends in Signal Processing*. Now, 2007, vol. 3, ch. 1, pp. 195–304.

[5] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69–88, 2002.

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[7] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech 2011*, 2011, pp. 437–440.

[8] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[9] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4520–4524.
[10] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 4277–4280.
[11] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.
[12] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: a neural network for large vocabulary conversational speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, 2016.
[13] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 396–409, 2017.
[14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239.
[15] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," Tech. Rep., 2017, arxiv.1706.05587.
[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.
[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 448–456, arxiv.1502.03167.
[18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," Microsoft Research, Tech. Rep., 2015, arxiv.1502.01852.
[19] A. K. Halberstadt, "Heterogeneous acoustic measurements and multiple classifiers for speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 1998.
[20] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.
[21] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," in *Proc. Interspeech 2016*, 2016, pp. 410–414.
[22] A. Mohamed, "Deep neural network acoustic models for ASR," Ph.D. dissertation, University of Toronto, 2014.
[23] A. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4148–4158.
[24] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Improving speech recognition by revising gated recurrent units," in *Proc. Interspeech 2017*, 2017, pp. 1308–1312.
[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.
[26] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.
[27] H. Sak, A. W. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Proc. Interspeech 2015*, 2015.
[28] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, pp. 749–753, 2018.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE