ON THE USE OF CONVOLUTIONAL NEURAL NETWORKS FOR SPEECH RECOGNITION

Borghini Alessia

Sapienza University of Rome, Italy

ABSTRACT

Automatic speech recognition (ASR) allows computers to acquire spoken human language by converting it into written text. In recent years, ASR has been applied in an increasing number of fields such as vocal assistance and customer service. Speech recognition models have reached impressive results by exploiting deep learning algorithm, and, more in detail, transformer architectures. However, the higher the complexity of the model, the more computational resources are needed to run experiments in a reasonable amount of time. Therefore, in this work we will compare two state-of-the-art architectures – both integrating Convolutional Neural Networks to optimize the number of model parameters – in a low-resource setting. In particular, we will train the smaller configurations of Conformer and ContextNet models, which uses a fewer number of hyper parameters, on a small portion of the LibriSpeech dataset. The comparison tests shown at the end of the paper are performed on the dev-clean, test-clean and test-other splits of the aforementioned dataset. This tests demonstrate that on the same amount of epochs Conformer is better than ContextNet from both performance and time perspectives.

Index Terms— Speech Recognition, RNN-Transducer, Transformer-Transducer, Convolutional Neural Networks.

1. INTRODUCTION

Automatic speech recognition, also referred to as speech-to-text, is the task of translating from spoken to written language. It is a key task that exploits knowledge from computer science, linguistics and computer engineering field. Speech recognition is applied in technologies such as Alexa[1] and Google Assistant[2] which are changing the way people interact with the their devices, homes, cars, and furthermore is important in Human Robot Interaction (HRI) field in order to simplify and make more natural the human-robot communication [1, 2, 3]. Additionally, it is closely related to the Natural Language Processing (NLP) area indeed, it is usually integrated into systems, such as Machine Translation systems, as a first step to obtain text to be processed or for multimodal purpose by adding vocal features.

Nowadays, in order to solve the speech recognition task, both hybrid and end-to-end (E2E) modeling approaches [4, 5] are adopted for different applications. Hybrid models [6, 7, 8, 9] combine Deep Neural Networks (DNNs) with Hidden Markov Models (HMMs). The neural part is involved for creating vectorized representations of the acoustic signals, while the HMM [10] is used to firstly learn the probabilities of the transitions between states. After that, a search graph is created combining the transitions probabilities with other knowledge sources, such as Language Models (LM), and then a decoder looks for the best solution over the obtained graph. End-to-end models [11, 12, 13] instead, map directly a input speech sequence into an output tokens sequence. These models use a single objective function for the optimization guaranteeing the achievement of a global optimum as opposed to hybrid models. Moreover, the hybrid models require more human effort in the building process, they are more complex computationally and consume more memory than the E2E counterpart. Thanks to their high performance, E2E models are catching on also in some industry applications [14, 15, 16]. However, hybrid models are still employed for commercial purpose, because in many cases features such as streaming, latency and adaptation capability are preferred over the performance.

Some widely used E2E architectures are: (1) Recurrent neural network-Transducer (RNN-T) [17, 18, 19, 20, 21], (2) Convolutional neural network [22, 23, 24, 25] and (3) Transformer-Transducer [26, 27, 28, 29, 30]. What emerges from the studies is that to achieve high performance in ASR task it is necessary to model long-range global context. Classic RNN-Transducer architecture use two RNNs [31, 32] in order to encode both the audio sequence and the textual output sequence and then combine the encoded vectors through a joint network. RNNs models well the temporal dependencies of variable length audio signals but are less able in capture the global context with respect to Transformer-based architectures. The Transformers [33], using the attention mechanism, allow the encoder to give greater importance to the most relevant part of the sequence. Their integration in RNN-T models, replacing the first RNN encoder, has lead to an increase of performance. CNN – despite has achieved

[1] https://developer.amazon.com/it-IT/alexa

[2] https://assistant.google.com/intl/it_it/

results close to the state-of-the-art [24, 25] – are more suitable for the extraction of fine-grained local feature patterns. In order to extend the context considered by the CNN a huge number of parameters is needed. Han et al. (2020) [34] proposed ContextNet, a CNN-based architecture in which they use Squeeze-and-Excitation layers [35] to squeeze the local features in single global embeddings. This strategy is able to keep low the number of parameters, however the obtained context is only a global average over the entire sequence. Some scientists merged together the extraction abilities of Transformers and CNNs to improve the performance of tasks such as question answering [36], machine translation [37], image classification and object detection [38]. Gulati et al. (2020) [39] instead studied how to combine convolution and self-attention for speech recognition in order to reduce the number of the model parameters needed, obtaining the Conformer architecture.

Speech recognition task needs a large amount of resources to conduct experiments which unfortunately not everyone has, thus in this work we will make a comparison of two state-of-the-art architectures in a low resource setting. In particular, we will compare Conformer and ContextNet architectures in their smallest configurations, i.e. both with about 10M hyper parameters, on a portion of the LibriSpeech dataset, the train-clean-100 split. We will focus on these two architectures because both integrates Convolutional Neural Networks to lower the number of parameters but in different fashion.

2. BACKGROUND

In this Section, we will introduce the architectures on which are based Conformer and ContextNet models. In particular, in Section 2.1, we will talk about RNN-Transducers, in Section 2.2 about Transformer-Transducers and in Section 2.3 about the Convolutional Neural Networks. Moreover, in Sections 2.4 and 2.5 we will briefly explain the loss function and the activation function used in the rest of the paper.

2.1. RNN-Transducers

Recurrent neural networks are widely used to map input sequences to output sequences but they require pre-alignment to do this. Furthermore, their sequential nature limits the possibility of parallelizing the computation during training. Graves (2012) [17] introduced the RNN-Transducer architecture which does not need pre-aligned data. This the model, indeed, is trained by minimize the RNN-T loss function which is defined as the logarithm of the sum of the probabilities of all possible alignments between text and audio sequences. The RNN-Transducer is composed by two RNNs – a transcription network and a prediction network – and a joint network. The transcription network encodes the audio input signal, while the prediction network encodes the textual target sequence based on the previous outputs. Then, the joint network predicts the tokens from the combination of the two encoded sequences which depends from the input sequence and the previous outputs.

2.2. Transformer-Transducers

Transformer-Transducers are an evolution of the RNN-Transducers where instead of the first RNN there is an attention module which performs the audio sequence modeling. In comparison with RNNs the attention mechanism may consider a longer context but it is also non-recurrent and therefore it can be easily parallelized.

Multi-head self attention (MHSA) [33] is a particular attention mechanism in which many independent attention are computed on an input sequence in parallel in order to capture dependencies of varying extension. Then, the resulting embeddings are concatenated and transformed with a linear layer to produce the final output. Moreover, the relative positional encoding can be used to allows the self-attention module to generalize better on different input length and to obtain a more robust encoder to the variance of the sequence length. Therefore, the relative positional encoding is an additional feature of the embeddings which gives information about the position in the sentence.

2.3. Convolutional Neural Networks

CNNs [40, 41] are commonly used for analyzing images because of their ability of capture spatial and temporal dependencies. This ability is due to the processing of the input which is not sequential but in blocks. These networks indeed apply filters, also called kernels, to 3-dimensional inputs in order to reduce them to a shape that is easier to process while maintaining the important features. The application of such kernels results in the computation of the convolution operation between the input and the kernel itself. The convolution operation is reported in the Equation 1

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n), \quad (1)$$

where I is the input and K is the kernel.

In recent CNNs architectures, skip connections [42, 43] are widely used for two main reasons: to avoid the gradient vanishing problem creating alternative paths for the gradient and to allow layers to learn from some information that are captured from the initial layers. Indeed, it was observed that the features learned by the firsts layers correspond to information with low semantic which are important for the following layers.

2.4. RNN-T Loss

RNN-T Loss sums the negative log probabilities of all the possible alignments. In particular, it computes the probability

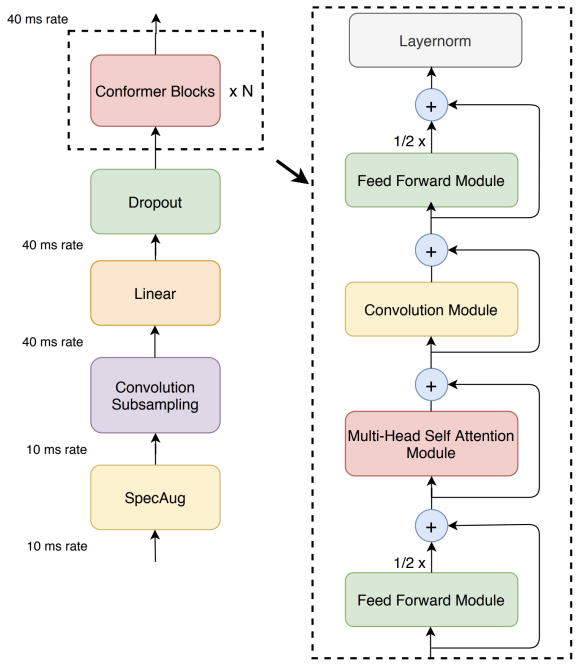

Fig. 1: Conformer encoder architecture

of a target label given the input one using the forward algorithm without considering the total alignment. Indeed, the obtained forward variable $\alpha(T_i, U_i)$ is the sum of probabilities for all paths ending at time-frame T_i and label position U_i. The loss is shown in Equation 2.

$$\text{RNN-T Loss} = -\sum_i logP(y_i|x_i) = -\sum_i \alpha(T_i, U_i). \quad (2)$$

2.5. Swish Activation Function

The swish activation function, introduced by Rramachandran et al. (2017) [44], is the best one for deep models always surpassing or equaling the ReLU. The equation is composed by the multiplication of x with a sigmoid function of βx, where β parameter is either constant or trainable. The swish function is shown in Equation 3.

$$Swish(x) = x \times sigmoid(\beta x) = \frac{x}{1 + exp(-\beta x)} \quad (3)$$

The function is equal to the sigmoid one for $\beta = 1$.

3. IMPLEMENTATION

The implementation of the models is done mainly with PyTorch[3] and torchaudio[4] libraries [45, 46]. For the implementation of the Conformer model we take inspiration from the

[3]https://pytorch.org/docs/stable/index.html
[4]https://pytorch.org/audio/stable/index.html

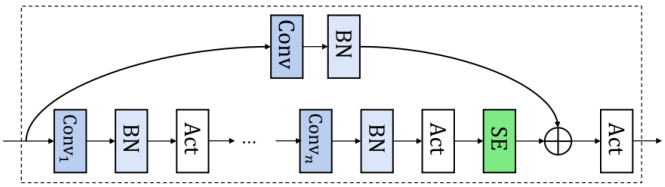

Fig. 2: Convolution block of ContextNet encoder

git repositories `https://github.com/sooftware/conformer` and `https://github.com/burchim/EfficientConformer`, while for the implementation of ContextNet from `https://github.com/upskyy/ContextNet`.

Both, Conformer and ContextNet papers, focus on the development of the encoding of the audio input sequences of a transducer architecture. Since the models uses attention mechanisms the resulting architectures are transformer-transducers.

In Section 3.1 Conformer and ContextNet encoders will be explained. In Sections 3.2 and 3.3, instead, respectively the decoder and the joint network will be presented.

3.1. Audio encoder

The audio encoder takes as input the audio sequence and outputs an encoded representation of the input. In order to obtain such embeddings, the audio signals are preprocessed. Using the short-time Fourier transformation over windowed segment the spectograms are obtained and then, they are converted to melspectograms through the mel scale. To makes the models generalize better data augmentation is performed by adding masks in frequency and time domain to the melspectograms. Now, the melspectograms are ready to be encoded with the one of the two encoders.

The Conformer encoder, as shown in Figure 1, is composed by a convolution subsampling layer and some conformer blocks. In particular, there are two sequential layers which apply the convolution, the batch normalization and the swish activation function to input, obtaining a multi-channel representation data. Then, the conformer blocks process the data with the following layers: (1) feed-forward module (FFN), (2) multi-head self attention (MHSA) module, (3) convolution module, (4) feed-forward module, and (5) a normalization layer. Two half-step FFNs are used at the beginning and the end of the conformer block as proposed by Lu et al. (2019) [47]. The MHSA module with relative positional encoding and dropout is applied to pre-normalized data. The convolutional module, instead, is characterized by pointwise convolution layers, the gated linear unit (GLU) and the depthwise convolution layer. The pointwise convolutional layer, for definition, iterates on every point of the

input changing the number of channels, while the depthwise one applies a different kernel to each different channel. The GLU activation function [48] controls the path through which information flows in the network avoiding the vanishing gradient. Finally, the layers are normalized [49] to reduce the training time.

The ContextNet encoder, instead, is based on the use of convolutional layers with skip connections and squeeze-and-excitation attention layers to capture the global context. This encoder is a sequence of 23 convolution blocks. Each block, as shown in Figure 2, is composed by a succession of convolution layers, batch normalization layers and swish activation functions. At the end of the block the squeeze-and-excitation layer and a skip connection with projection are applied to the output. The squeeze-and-excitation (SE) layer is used to compress the local features in global embeddings allowing CNNs to have access to a wider context. This attention mechanism improves the generalization capabilities through the use of three main components: (1) the squeeze module, (2) the excitation module, and (3) the scale module. The first module is an average pooling layer which reduces the spatial dimensions of the feature maps to a single value. Then, the excitation module is a bottleneck formed by two fully connected layers, the first compresses the dimension and the second decompresses it, thus obtaining an output of the same dimension of the input. The final module, the scale one, applies the sigmoid activation function to the output scaling its values to a range of (0,1). Finally, the output is multiplied element-wise with the input of the SE module.

3.2. Text decoder

The text decoder takes as input the target text sequence and returns a contextualized representation of such sequence. Both models, uses as decoder a single layer unidirectional LSTM network [50] followed by a linear layer.

3.3. Joint network

Once the audio and the text sequences have been encoded, they are processed by a joint network to produce a prediction which will be decoded by the tokenizer to obtain the final sentence. The joint network concatenates the two embeddings and applies a linear layer and a softmax layer to obtain a probability distribution of the final labels. These labels are the sub words which will compose the predicted sentence.

4. EXPERIMENTAL SETUP

In this section we will provide details on the experiments we have done. In Section 4.1 we talk about the dataset used for the experiments and about the preprocessing. Then, in Sections 4.2 and 4.3 we explain how the training and the evaluation phases are structured.

	# samples	Conformer	ContextNet
train-100h	28539	90 min/epoch	140 min/epoch
eval	2703	7 min/epoch	8 min/epoch
test-clean	2620	7 min/epoch	8 min/epoch
test-other	2939	7 min/epoch	8 min/epoch

Table 1: Conformer and ContextNet models computation times with Tesla K80 GPU

Model	Conformer	ContextNet
Encoder dim	144	640
Encoder layers	16	5
Attention heads	4	-
Decoder dim	320	640
Decoder layers	1	1
Convolutional kernel size	31	5
Dropout	0.1	0.3
Num Params (M)	10.3	10.8

Table 2: Conformer and ContextNet models parameters

4.1. Data

We evaluate the models on the publicly available LibriSpeech ASR corpus [51]. This dataset contains 970 hours of speech data with corresponding text transcription and an additional textual corpus used mainly for the training of the language models. Due to the unavailability of powerful computation resources, the models have been trained on the train-clean-100 split, a portion of the LibriSpeech dataset containing 100 hours of audio data with the corresponding transcripts. In Table 1, the dimension of this dataset split and the other used then for the test are provided along with the computation times taken from the two models on Tesla K80 GPU (12 GB) for training and testing. The time needed by the models for an epoch on the training set, the evaluation times and the testing times are reported.

Before the training phase begins, we tokenize the transcript data using the sentencepiece tokenizer[5] on sub words and with a vocab size of 256. Then, the model extracts from the audio signals with Spectogram and MelSpectogram[6] transformers 80 dimensional filter banks features using a 25ms window with a stride of 10 ms. Only during the training phase, also SpecAugment has been used [52, 53] for masking the spectrogram in the frequency domain with a maximum frequency of $F = 27$, and then in the time domain with a maximum length of the mask proportional to $pS = 0.05\%$.

[5] https://github.com/google/sentencepiece

[6] https://pytorch.org/audio/stable/transforms.html

4.2. Training

When the model ends the forward process we compute the training loss between the sequence predicted by the model and the target sequence using the CUDA-warp RNN-T Loss[7]. We train the models with Adam optimizer [54] using as exponential decay rate for the moments estimation $\beta_{1,2} = (0.9, 0.98)$ and a small epsilon to avoid division by zero $\epsilon = 10^{-8}$. We also add $l_2 = 10^{-6}$ penalty for regularization. To control the learning rate of the optimizer we use the transformer learning rate scheduler [33] with $10k$ warm-up steps and, for Conformer a peak of $0.005/sqrt(d)$ (where $d = 144$ is the dimension of the model) while for ContextNet a peak of 0.0025. Automatic Mixed Precision Training with Gradient Accumulation[8] has been used to reduce memory requirements and to speed up the computation while maintaining high accuracy. In particular, the Autocast chooses the best precision for GPU to improve the performance and the GrandScaler avoid gradient underflow. The gradient accumulator, instead, compensates the lack of memory, increasing the batch size, and speed up the training phase reducing the amount of the network weights updates.

We train the small configurations of both Conformer and ContextNet following the indications provided by the authors. The hyper-parameters of the models are shown in Table 2. For regularization, both uses dropout but with a different probability of turn off the neurons. The dimensions of the encoder and the decoder are also very different, the smaller dimensions of the Conformer model are compensated by the use of the attention heads. The total number of parameters for the small configurations indeed is almost the same.

4.3. Evaluation

For evaluating the models, we decode the input sequences using the greedy search decoder algorithm which is shown in Listing 1. This algorithm encodes the input audio sequence, creates an encoded representation of the tokens predicted so far and the current hidden states with the decoder and predicts the next token with the joint network. The joint network produces the logits from which the probability distribution of the tokens from the vocabulary is computed. From this probability the most probable value is extracted and, using the tokenizer, the corresponding token is obtained. When a new token is predicted the list of the predicted tokens so far is updated and it is decoded obtaining `g`. On the one hand, the same encoded part of the audio sequence, `f`, can be used for different decoded text sequences until it produces a the "null" token – which corresponds to the value `0` – or the maximum number of consecutive decoding is reached. On the other hand, each decoded text sequence can be combined with the subsequent encoded parts of the audio sequence if the predicted token is "null".

Once the greedy search algorithm has finished, the obtained sequence prediction is compared with the target one using the word error rate[9] (WER) [55] metric which is defined as follows:

$$\text{WER} = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C},$$

where S is the number of substitutions, D the number of eliminations, I the number of insertion, C the number of correct words and N is the number of words in the reference.

	Conformer	ContextNet
WER test-clean	**33.26**	63.66
WER test-other	**53.04**	78.77
WER dev-clean	**32.21**	64.47

Table 3: Conformer and ContextNet WER on LibriSpeech

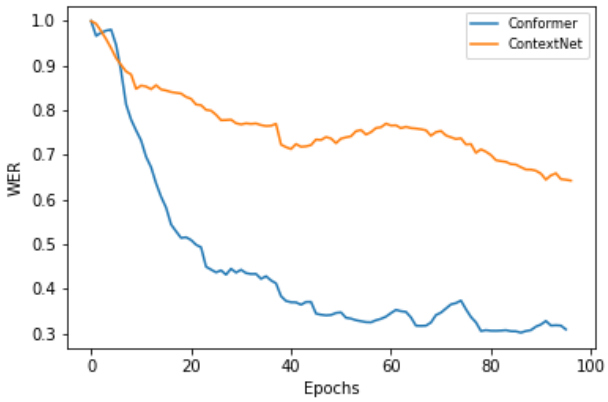

Fig. 3: WER plot over epochs of Conformer and ContextNet

5. RESULTS

In this Section the results obtained in the experiments described above are reported. In particular, in Section 5.1 are discussed the quantitative results, while in Section 5.2 the qualitative ones.

5.1. Quantitative results

Table 3 compares the WER performance of Conformer and ContextNet models. All results are rounded up to 2 digits after the decimal point. The Figure 3 shows the evolution of the WER on the clean evaluation set of the two models. The Conformer model converges faster, in fact with the same number of epochs it achieves higher performance than ContextNet

[7] https://github.com/lytic/warp-rnnt

[8] https://pytorch.org/docs/stable/notes/amp_examples.html#gradient-accumulation

[9] https://github.com/jitsi/jiwer

Listing 1 Greedy search decoding algorithm

```
def greedy_search_decoding(x, x_len):
    preds = []
    f, f_len = model.encoder(x, x_len)

    for b in range(x.size(0)):

        y = x.new_zeros(1, 1, dtype=torch.long)
        hidden = None

        enc_step = 0
        consec_dec_step = 0

        while enc_step < f_len[b]:

            g, hidden = model.decoder(y[:, -1:], hidden_states=hidden)

            while enc_step < f_len[b]:

                logits = model.joint(f[b:b+1, enc_step], g[:, 0])
                pred = logits.softmax(dim=-1).log().argmax(dim=-1)

                if pred == 0 or consec_dec_step == 5:
                    consec_dec_step = 0
                    enc_step += 1
                else:
                    consec_dec_step += 1
                    y = torch.cat([y, pred.unsqueeze(0)], axis=-1)
                    break

        preds += tokenizer.decode(y[:, 1:].tolist())

    return preds
```

which requires more training epochs to learn. The results on the test-clean are consistent with the results obtained on the dev-clean set used as evaluation for the training phase of the models. As expected the errors on the test-clean, instead, are lower than the ones on the test with noise, the test-other. The increase of error with ContextNet model is lower with respect to Conformer model.

5.2. Qualitative results

To understand the mistakes that models make and how the performance are translated in a more practical way, we select a sample of sentences from the test-clean and test-other datasets with the corresponding model predictions. The sentences are reported in Table 4 with the relative word error rates. As expected Conformer predicts more accurate sentences than ContextNet, but in both cases the error rates are highly variable depending on the input sequence. The errors made by Conformer are usually about the prediction of words pronounced similarly to the target ones, while ContextNet instead, sometimes does not even predicts the words that correspond to some part of the audio sequence or predicts words that does not fully match the pronunciation. In both cases almost all the predicted words have a meaning taken individually, but when we consider the context they make less sense. For this reason, using language models in the decoding phase – instead of the greedy search decoding strategy – could significantly improve the performances. A language model, given sequence of words, computes the probability that any given word is the next one mitigating the misunderstanding due to the pronunciation of the words. Additionally, we have used a tokenizer with a vocabulary size of 256 sub tokens. With a bigger vocabulary, the tokenized text sequences will be on average shorter and each element will have a larger range value. This could allows an increase of performance on one hand but, on the other hand the slowing down the model convergence. It is also possible to pre-train with text-only data the decoder of the models in order to improve the quality of such embeddings.

	Pred/Ground	Sentence	WER
ContextNet	Prediction	***he*** *left him then for the* ***journey*** *to* ***all*** *the door and* ***screwed him*** *to the office*	0.333
	Groundtruth	***they*** *left him then for the* ***jailer arrived*** *to* ***unlock*** *the door and* ***escort them*** *to the office*	
	Prediction	*meanwhile he had called upon me to make a report of the three* ***sister*** *in* ***ending*** *as the* ***head consumed that joints*** *and mister being independent* ***and venus property*** *at the same time*	0.553
	Groundtruth	*meanwhile he had called upon me to make a report of the three* ***wire system known*** *in* ***england*** *as the* ***hopkinson both doctor john hopkinson*** *and mister* ***edison*** *being independent* ***inventors*** *at* ***practically*** *the same time*	
	Prediction	*the* ***other depict*** *which existed* ***forese on the mariposa*** *to* ***serve the beneister the*** *man who looked after* ***him*** *for* ***two*** *years* ***in his own astonish absurd about*** *suspicion*	0.389
	Groundtruth	*the* ***only duplicate*** *which existed* ***so far as i knew was that which belonged*** *to* ***my servant bannister a*** *man who* ***has*** *looked after* ***my room*** *for* ***ten*** *years* ***and whose honesty is absolutely above*** *suspicion*	
	Prediction	***david did*** *for the* ***lords*** *of the prisoners but* ***for*** *the* ***great explanation*** *the* ***greater the count***	0.591*
	Groundtruth	***daphne again pleaded*** *for the* ***liberation*** *of the prisoners but* ***philippus silenced her with*** *the* ***grave exclamation*** *the* ***order of the king***	
Conformer	Prediction	*the most* ***vanis*** *of* ***the mole*** *was the overthrow of the island of a* ***twantous***	0.357
	Groundtruth	*the most* ***famous*** *of* ***them all*** *was the overthrow of the island of* ***atlantis***	
	Prediction	*there was a bright moonlight* ***broke into my*** *shadows of overhanging and* ***ritale*** *leaves and the* ***modle*** *lights and shadows glided* ***outly*** *across his* ***pill*** *features*	0.296
	Groundtruth	*there was a bright moonlight* ***broken by the*** *shadows of overhanging boughs and* ***withered*** *leaves and the* ***mottled*** *lights and shadows glided* ***oddly*** *across his* ***pale*** *features*	
	Prediction	*there was something in his air and* ***the*** *manner that betrayed to the* ***skout*** *the utter confusion of the state of his mind*	0.091
	Groundtruth	*there was something in his air and manner that betrayed to the* ***scout*** *the utter confusion of the state of his mind*	
	Prediction	*it wouldn't do you* ***in all*** *after that story came out* ***from*** *me* ***in*** *the* ***bless trays were he said*** *in the case as well as* ***all the divesing numbers*** *of the bar to be seen there* ***he thinks blind*** *to the* ***cur***	0.415*
	Groundtruth	*it wouldn't do you* ***know*** *after that story came out* ***for*** *me* ***and*** *the* ***vice chancellor who sat*** *in the case as well as* ***other judges and members*** *of the bar to be seen there* ***kenneth explained*** *to the* ***colonel***	

Table 4: Examples of predicted sentences made by Conformer and ContextNet models compared with the target sentences and with the corresponding WER. The sentences with the '*' symbol are taken from the test-other split of LibrSpeech dataset, the others instead from the test-clean split of LibrSpeech dataset.

6. CONCLUSIONS

In this work, we compared two models which incorporates in different way the CNNs for end-to-end automatic speech recognition. In Conformer the CNNs are used to improve an attention-based architecture, ContextNet instead, is an architecture based on CNNs and squeeze-and-excitation module. The experiments were done in a low data setting, the train were done on the train-clean-100 split of LibriSpeech dataset, while the tests on the whole test sets, both test-clean and test-other. Our results demonstrate that Conformer achieves better performance than ContextNet with the same number of epochs.

7. REFERENCES

[1] Rainer Stiefelhagen, Christian Fugen, R Gieselmann, Hartwig Holzapfel, Kai Nickel, and Alex Waibel, "Natural human-robot interaction using speech, head pose and gestures," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. IEEE, 2004, vol. 3, pp. 2422–2427.

[2] Raveesh Meena, Kristiina Jokinen, and Graham Wilcock, "Integration of gestures and speech in human-robot interaction," in *2012 IEEE 3rd International*

Conference on Cognitive Infocommunications (CogInfoCom). IEEE, 2012, pp. 673–678.

[3] Javier G Rázuri, David Sundgren, Rahim Rahmani, Antonio Moran, Isis Bonet, and Aron Larsson, "Speech emotion recognition in emotional feedbackfor human-robot interaction," *International Journal of Advanced Research in Artificial Intelligence (IJARAI)*, vol. 4, no. 2, pp. 20–27, 2015.

[4] Rohit Prabhavalkar, Kanishka Rao, Tara N Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly, "A comparison of sequence-to-sequence models for speech recognition.," in *Interspeech*, 2017, pp. 939–943.

[5] Juan M Perero-Codosero, Fernando M Espinoza-Cuadros, and Luis A Hernández-Gómez, "A comparison of hybrid and end-to-end asr systems for the iberspeech-rtve 2020 speech-to-text transcription challenge," *Applied Sciences*, vol. 12, no. 2, pp. 903, 2022.

[6] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "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] Herve A Bourlard and Nelson Morgan, *Connectionist speech recognition: a hybrid approach*, vol. 247, Springer Science & Business Media, 2012.

[8] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

[9] Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al., "Transformer-based acoustic modeling for hybrid speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6874–6878.

[10] Mark Gales and Steve Young, *The application of hidden Markov models in speech recognition*, Now Publishers Inc, 2008.

[11] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, "Wav2letter: an end-to-end convnet-based speech recognition system," *arXiv preprint arXiv:1609.03193*, 2016.

[12] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[13] Jinyu Li, "Recent advances in end-to-end automatic speech recognition," *arXiv preprint arXiv:2111.01690*, 2021.

[14] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al., "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[15] Tara N Sainath, Yanzhang He, Bo Li, Arun Narayanan, Ruoming Pang, Antoine Bruguier, Shuo-yiin Chang, Wei Li, Raziel Alvarez, Zhifeng Chen, et al., "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6059–6063.

[16] Jinyu Li, Rui Zhao, Zhong Meng, Yanqing Liu, Wenning Wei, Sarangarajan Parthasarathy, Vadim Mazalov, Zhenghao Wang, Lei He, Sheng Zhao, et al., "Developing rnn-t models surpassing high-performance hybrid models with customization capability," *arXiv preprint arXiv:2007.15188*, 2020.

[17] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[18] Eric Battenberg, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu, "Exploring neural transducers for end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 206–213.

[19] Linhao Dong, Shuang Xu, and Bo Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[20] George Saon, Zoltán Tüske, Daniel Bolanos, and Brian Kingsbury, "Advancing rnn transducer technology for

speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5654–5658.

[21] Surabhi Punjabi, Harish Arsikere, Zeynab Raeesy, Chander Chandak, Nikhil Bhave, Ankish Bansal, Markus Müller, Sergio Murillo, Ariya Rastrow, Andreas Stolcke, et al., "Joint asr and language identification using rnn-t: An efficient approach to dynamic language switching," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7218–7222.

[22] Yu Zhang, William Chan, and Navdeep Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4845–4849.

[23] Neil Zeghidour, Qiantong Xu, Vitaliy Liptchinsky, Nicolas Usunier, Gabriel Synnaeve, and Ronan Collobert, "Fully convolutional speech recognition," *arXiv preprint arXiv:1812.06864*, 2018.

[24] Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M Cohen, Huyen Nguyen, and Ravi Teja Gadde, "Jasper: An end-to-end convolutional neural acoustic model," *arXiv preprint arXiv:1904.03288*, 2019.

[25] Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6124–6128.

[26] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.

[27] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[28] Ching-Feng Yeh, Jay Mahadeokar, Kaustubh Kalgaonkar, Yongqiang Wang, Duc Le, Mahaveer Jain, Kjell Schubert, Christian Fuegen, and Michael L Seltzer, "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[29] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[30] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7829–7833.

[31] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[32] John J Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[34] Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu, "ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context," in *Proc. Interspeech 2020*, 2020, pp. 3610–3614.

[35] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[36] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," *arXiv preprint arXiv:1804.09541*, 2018.

[37] Baosong Yang, Longyue Wang, Derek Wong, Lidia S Chao, and Zhaopeng Tu, "Convolutional self-attention networks," *arXiv preprint arXiv:1904.03107*, 2019.

[38] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE/CVF*

international conference on computer vision, 2019, pp. 3286–3295.

[39] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," *Proc. Interspeech 2020*, pp. 5036–5040, 2020.

[40] Kunihiko Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, no. 2, pp. 119–130, 1988.

[41] Yann LeCun, Yoshua Bengio, et al., "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] Sasha Targ, Diogo Almeida, and Kevin Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[44] Prajit Ramachandran, Barret Zoph, and Quoc V Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., pp. 8024–8035. Curran Associates, Inc., 2019.

[46] Yao-Yuan Yang, Moto Hira, Zhaoheng Ni, Anjali Chourdia, Artyom Astafurov, Caroline Chen, Ching-Feng Yeh, Christian Puhrsch, David Pollack, Dmitriy Genzel, Donny Greenberg, Edward Z. Yang, Jason Lian, Jay Mahadeokar, Jeff Hwang, Ji Chen, Peter Goldsborough, Prabhat Roy, Sean Narenthiran, Shinji Watanabe, Soumith Chintala, Vincent Quenneville-Bélair, and Yangyang Shi, "Torchaudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[47] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu, "Understanding and improving transformer from a multi-particle dynamic system point of view," *arXiv preprint arXiv:1906.02762*, 2019.

[48] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[49] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[50] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[51] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[52] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[53] Daniel S Park, Yu Zhang, Chung-Cheng Chiu, Youzheng Chen, Bo Li, William Chan, Quoc V Le, and Yonghui Wu, "Specaugment on large scale datasets," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6879–6883.

[54] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[55] Andrew Cameron Morris, Viktoria Maier, and Phil Green, "From wer and ril to mer and wil: improved evaluation measures for connected speech recognition," in *Eighth International Conference on Spoken Language Processing*, 2004.